University of Alberta

Determining Structure in Test Performance:

An Artificial Neural Network Approach

by

Gregory Steven Sadesky

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Measurement, Evaluation, and Cognition

Department of Educational Psychology

Edmonton, Alberta

Spring 2007

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-29733-9
Our file Notre référence
ISBN: 978-0-494-29733-9

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Dedication

I dedicate this work to my wonderful wife and friend Susan Cairns, though I know she won't much want it; how a Voronoi tessellation might approximate the Bayesian decision boundary isn't really her cup of tea. Nevertheless, she is directly responsible for enabling me to be inspired by such things, by believing in me, sacrificing for me, and perhaps most remarkably, by patiently listening to me as I tried to articulate the revelation *du jour*. How I have reached this point in my life and my understanding requires no further research; I stood on her shoulders.

Abstract

The Kohonen Self-Organizing Map (SOM), a kind of artificial neural network, is evaluated for its efficacy in determining test structure in educational measurement applications. It is argued that the SOM may be particularly useful for this function since it can reveal both the dimensional (latent trait) and class (latent state) structure of complex data. A series of monte carlo experiments assessed the capacity of one- and two-dimensional, small and large SOMs to determine the structure of data composed of dichotomously-scored test items. These data were simulated to comprise latent classes and varied with respect to the discrimination of the individual items and the dimensionality of the data as a whole. In addition to the important role for item discrimination in producing high quality projections and low quantization error, the relationship between characteristics of the map and the complexity of the data was found to be critical for the SOM to effectively represent test data. In particular, it was determined that SOMs most accurately preserved adjacency and proximity relationships when the intrinsic dimensionality of the data matched the number of co-ordinate axes of the map. Implications for future applications of SOMs in educational measurement are discussed, as well as suggestions for further research.

Acknowledgements

First and foremost, I would like to express my deep gratitude to my co-supervisors, Jackie Leighton and Mike Dawson. Thank you for providing me the right balance of structure and autonomy, rigour and creativity, and of course, psychology and measurement. I thrived intellectually under your supervision and I know I've been very fortunate to have worked with you both. I also owe a debt of gratitude to Mike Carbonaro who, in my first semester as a doctoral student, showed me that Education and connectionism could be a natural fit. To my student colleagues, especially Liam, Matthew, Rebecca, Katie, Nizam, and Marilyn, but many others, thanks first for your friendship, and for many amazing discussions, both in and out of the lab. Also, I want to thank Todd Rogers and Mark Gierl. Todd, you amaze me with your total commitment to your students and to Educational Measurement. And Mark, it seems like you've always been there when I've needed career advice, guidance and support. Without you, I wouldn't be in (and out of) CRAME. Last, I want to say how much both the influence and support of my family has meant to me in this accomplishment. Your contribution extends far beyond what I can articulate, but here are a few biggies. You've taught me to believe in myself and in what I can accomplish, to value (and love) education, you've modeled for me love, persistence, patience, respect, and success. You are with me always.

Table of Contents

Chapter 1 - Introduction 1

References 7

Chapter 2 – Methods to Determine Test Structure 13

Overview 13

Dimensional-Based Methods for Determining Test Structure 13

Factor Analysis 16

Multidimensional Scaling 21

DETECT 25

Summary 28

References 29

Chapter 3 – Latent Class Accounts of Test Structure 33

Class-Based Accounts for Determining Test Structure 35

Cluster Analysis 35

Latent Class Analysis 40

Summary 52

References 54

Chapter 4 – Artificial Neural Networks 57

What are ANNs? 58

Supervised ANNs 61

Unsupervised ANNs 62

Kohonen's SOM 63

Applications of SOMs in Educational Measurement 69

References 76

Chapter 5 - Experiment One: SOM Representation of Ordered Classes 79

Method 80

Training the SOM 83

Analysis of SOM Performance 87

Map-Data Fit: Quantization Error 88

Projection 92

Results 96

Section I – Statistical Results 96

Section II – Qualitative Examination of the SOM 100

Section III – Interpretation of the SOM 107

General Discussion 111

References 114

Chapter 6 - Experiment 2: SOM Representation of Classes Ordered in Two Dimensions 115

Method 116

Results 116

Section I – Statistical Analysis 117

Section II – Qualitative Examination of the SOM 121

Section III – Interpretation of the SOM 125

General Discussion 128

Chapter Seven – Experiments 3 and 4: One-Dimensional SOMs 131

Experiment 3: Data and Method 132

Results 132

Section I – Statistical Analysis 132

Section II – Qualitative Examination of SOMs 135

Section III – Interpretation of the SOM 137

Experiment 4: Data and Method 139

Section I – Statistical Analysis 140

Section II – Qualitative Examination of SOMs 142

Section III – Interpretation of the SOM 144

Discussion – Experiments 3 and 4 146

Combined Analysis of All Experiments 147

Quantization Error 147

Topological Preservation 149

Correlation of Distances 151

Discussion – Combined Analysis 153

Chapter 8 – General Discussion 154

Using the SOM to Determine Test Structure - Considerations 154

Characteristics of Data 154

Characteristics of the SOM 157

Dimensional Match between Data and Map 159

SOMs as statistical models 161

An Evaluation of a Current Application of SOMs in Educational Measurement 168

Conclusion 172

References 173

Appendix A – Data Generator: User Form and Visual Basic Code 175

Visual Basic Code 176

Code for Form Controls 176

Data_Generator module 178

Appendix B – SOM Engine: User Form and Visual Basic Code 182

Visual Basic Code 183

Code for Form Controls 183

SOM Engine Module 185

List of Tables

Table 3.2. Manifest Joint Probabilities for 3 Dichotomous Items 42

Table 3.3. Latent Parameters Leading to Manifest Joint Probabilities in Table 3.2. 43

Table 3.5. Observed data presented in Goodman (1974). 47

Table 3.6. Class Probabilities and Class Conditional Probabilities for Each Item for both Restricted and Unrestricted Two Latent Class Models 49

Table 4.1. Interpreting Model Vectors from a 2 x 2 SOM 65

Table 5.1. Characteristics of the Training, Data, and the Self-Organizing Map for Experiment One. 85

Table 5.2. Mean (Standard Error) Quantization Error by Condition 97

Table 5.3. Mean (Standard Error) Topological Preservation by Condition 98

Table 5.4. Mean (Standard Error) Correlations (R_{dist}) between Distances in Metric Space and Associated SOM Co-ordinates by Condition 99

Table 5.5. Two Sample Model Vectors 101

Table 5.6. Mean (Standard Deviation) Number of Matches between Intended Class and Modal Class by Condition (Maximum = 500) 105

Table 5.7. Stress1 and Variance Accounted For (VAF) in One- and Two-Dimensional MDS Analyses on Typical Replications in Each Condition 108

Table 5.8. Correlation between Dimensional Co-ordinates of Each SOM Unit in MDS Analyses and Total Expected Score for Unit Members, Performed Separately for Typical Replications of Each Condition 109

Table 5.9. Correlation between Dimensional Co-ordinates and Unit Conditional Item Probabilities from Ten Replications of Condition One. 110

Table 6.1. Mean (Standard Error) Quantization Error by Condition 117

Table 6.2. Mean (Standard Error) Topological Preservation by Condition 118

Table 6.3. Mean (Standard Error) Correlations (R_{dist}) between Distances in Metric Space and Associated SOM Co-ordinates by Condition 119

Table 6.4. Stress1 and Variance Accounted For (VAF) in One- and Two-Dimensional MDS Analyses on Typical Replications in Each Condition 126

Table 6.5. Correlation between Dimensional Co-ordinates of Each SOM Unit in MDS Analyses and Total Expected Score for Unit Members, Performed Separately for Typical Replications of Each Condition 127

Table 7.1. Mean (Standard Error) Quantization Error by Condition 133

Table 7.2. Mean (Standard Error) Topological Preservation by Condition 134

Table 7.3. Mean (Standard Error) Correlations (R_{dist}) between Distances in Metric Space and Associated SOM Co-ordinates by Condition 134

Table 7.4. Stress1 and Proportion of Variance Accounted For (VAF) in One- and Two-Dimensional MDS Analyses on Typical Replications in Each Condition 137

Table 7.5. Correlation between Dimensional Co-ordinates of Each SOM Unit in MDS Analyses and Total Expected Score for Unit Members, Performed Separately for Typical Replications of Each Condition 139

Table 7.6. Mean (Standard Error) Quantization Error by Condition 140

Table 7.7. Mean (Standard Error) Topological Preservation by Condition 141

Table 7.9. Stress1 and Proportion of Variance Accounted For (VAF) in One- and Two-Dimensional MDS Analyses on the Model Vectors from Typical Replications in Each Condition in Experiment 4 144

Table 7.10. Correlation between MDS Dimensional Co-ordinates of Each SOM Unit and Total Expected Score by Subscale for Unit Members, Performed Separately for Typical Replications of Each Condition in Experiment 4 145

Table 7.11. Mean Quantization Error across All Conditions by Map Size, Item Discrimination, and Dimensional Match 147

Table 7.12. Mean Topological Preservation Across All Conditions by Map Size, Item Discrimination, and Dimensional Match 149

Table 8.1. KR-20 reliabilities for each unidimensional dataset. 156

List of Figures

Figure 1.1. A Possible Relationship between Item-level Performance and Test Summary Measure in a One-Dimensional Test 2

Figure 1.2. The Relationship between Item-level Performance and Test Summary Measures in a Two-Dimensional Test 3

Figure 2.1. Unrotated versus rotated factor solutions 18

Figure 3.1. The Shape of Mixture Distributions According to Degree of Overlap of the Underlying Class Distributions 35

Figure 4.1. Neural Network Representation of a Linear Regression Equation 59

Figure 4.2. Three Layer Artificial Neural Network 60

Figure 4.3. Architecture of the Kohonen Self-Organizing Map 64

Figure 4.4. Two Neighbourhoods Defined on a SOM. 68

Figure 5.1. SOM Representation of Probability Density. 89

Figure 5.2. The Position of Model Vectors and Resulting Quantization Error 91

Figure 5.3. Projection Quality of Model Vectors in the Metric Space as a Function of the Placement of Corresponding Cb-ordinates in the Self-Organizing Map 94

Figure 5.2. Most Frequent Intended Class Membership for Two Self-Organizing Maps in Experiment One (Condition = Small Map, Item Discrimination 2.0) 102

Figure 5.3. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment One, Small Map Conditions 104

Figure 5.4. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment One, Large Map Conditions 106

Figure 6.1. Modal Simulated Class Membership for Two Self-Organizing Maps in Experiment Two, Small Map, Item Discrimination = 2.0 122

Figure 6.3. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Two, Conditions Four, Five, and Six. 124

Figure 7.1. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Three, Small Maps Only 135

Figure 7.2. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Three, Large Maps Only 136

Figure 7.3. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Four, Conditions One, Two, and Three. 143

Figure 7.4. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Three, Conditions Four, Five, and Six. 143

Figure 7.5. Mean Quantization Error by Item Discrimination and Dimensional Match for All Small Map Conditions. 148

Figure 7.6. Mean Quantization Error by Item Discrimination and Dimensional Match for All Large Map Conditions. 148

Figure 7.7. Mean Topological Preservation by Item Discrimination and Dimensional Match for All Small Map Conditions 150

Figure 7.8. Mean Topological Preservation by Item Discrimination and Dimensional Match for All Large Map Conditions 150

Figure 7.10. Mean Correlation of Distances by Item Discrimination and Dimensional Match for All Large Map Conditions 152

Figure 8.1. Representation of performance states derived from self-organizing neural network in Stevens, Johnson, & Soller (2005) 170

Figure A1. Form to Specify Data Generation Parameters 175

Figure B1. Form to Set SOM Parameters 182

Determining Structure in Test Performance – An Artificial Neural Network Approach

Chapter 1 - Introduction

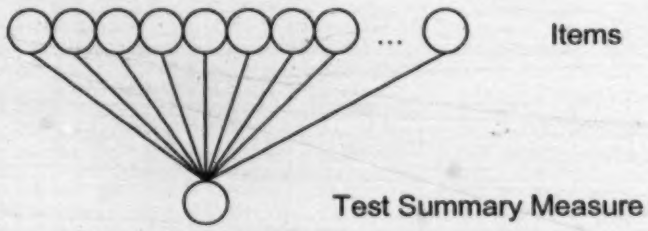
One of the essential functions of educational measurement is to summarize complex data. In particular, the data comprising raw or scored student responses must be transformed into summary measures that make student performance overall more transparent and interpretable. Because these data usually comprise many measures (e.g., responses to test items), such summaries are essential to facilitate a meaningful interpretation of performance.

Summaries of student data are not only to make data appear more simple in structure; simplification follows from the structure of the data itself. For most tests, performance on large numbers of items can be well described by a small number of variables. Consider a hypothetical achievement test comprising multiple-choice items, represented in Figure 1. The scored student responses to this test consist of *n*-dimensional vectors, or alternatively, points in an *n*-dimensional space. Performance on these items is represented in Figure 1 by the first row of circles. Interpretation of student performance based on all item responses simultaneously is unmanageable and therefore, some summary of performance has to be found. But which summary should be chosen as the best representation of performance on the test as a whole?

To effectively address this question, the pervasive relationships among responses to the test items, known as *test structure*, must be carefully examined. When this examination reveals a consistent structure, a test summary that follows that structure can be adopted. For example, a situation common to many standardized tests is that responses to all items are positively related; they share common variance. In this case, a

single measure may be a good summary of overall test performance. This is represented in Figure 1.1 by the connection of all items to a single circle, representing the strength of a single dimension underlying performance on all items and thus the reasonableness of adopting a single measure as a test summary. For other tests, responses to certain test items might be positively related only to certain other test items, depicted in Figure 1.2. In this case, an appropriate summary of test performance could be two separate measures, represented by the two circles to which the subsets of items connect.

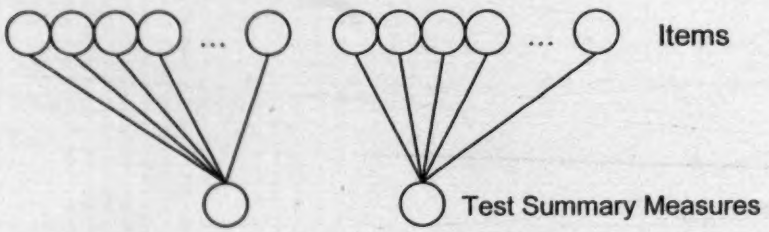
Figure 1.1. A Possible Relationship between Item-level Performance and Test Summary Measure in a One-Dimensional Test



The correct identification of test structure is an essential step in creating test summaries that can be validly interpreted. Numerous studies highlight the errors that result from misidentifying test structure for the purposes of summarizing student performance (e.g., Sireci, Thissen, & Wainer, 1991; Tate, 2004; Walker & Beretvas, 2003; Gitomer & Yamamoto, 1991; Zenisky, Hambleton, & Sireci, 2002), and also for estimating certain item (e.g., IRT parameters) and test parameters (e.g., reliability). In addition, certain test analysis procedures depend upon the correct identification of the dimensionality of the test, including those associated with parameter estimation in item

response theory (IRT) such as BILOG and LOGIST as well as procedures for detecting differential item functioning, such as SIBTEST.

Figure 1.2. The Relationship between Item-level Performance and Test Summary Measures in a Two-Dimensional Test



Given the crucial role of test structure in test interpretation and the consequences resulting from its incorrect specification, it is no surprise that many measures exist for its determination. In large scale standardized testing contexts, these methods are often focused on an assessment of the appropriateness of a unidimensional model for test performance. These methods include factor analysis of tetrachoric correlations (e.g., Gessaroli & de Champlaim, 1986; Hambleton & Rovenelli, 1986; McDonald, 1967, 1982; Mislevy, 1987), tests of local independence (e.g., Hattie, 1985; Nandakumar, 1994; Stout, 1987; Stout, Habing, Douglas, Kim, Roussos, & Zhang, 1996; Zhang & Stout, 1999), and even traditional measures of reliability such as KR-20 (Kuder & Richardson, 1937). These methods are often employed to garner support for the use of traditional unidimensional scoring models such as those described for unidimensional item response theory, or the total score model advanced by classical test theory.

Though conceptually fundamental to test quality, the above-described unidimensional models can not adequately characterize test structure for many contemporary educational measurement applications. Inferences about examinee achievement are desired that go beyond the traditional standardized score or percent correct (e.g., Gitomer & Yamamoto, 1991; Mislevy, 1996; Mislevy, Almond, Yan, & Steinberg, 2000; Tatsuoka, 1983, 1990, 1995; Yamamoto, 1987; Yamamoto & Gitomer, 1993), more complex often interactive tasks are being used in assessment contexts (Stevens, Johnson, & Soller, 2005; Stevens, Soller, Cooper, & Sprang, 2004; Stevens & Palacio-Cateyano, 2003; Stevens, Ikeda, Casillas, Palacio-Cayetano, & Clyman, 1999) and computers being used to implement automated scoring algorithms (Williamson, Bejar, & Sax, 2004; Rudner & Liang, 2002; Williamson & Bejar, 2000; Williamson, Bejar, & Hone, 1999). The valid interpretation of performance for all of these new applications requires the accurate specification of test structure. Methods to determine test structure developed under the assumptions of unidimensionality of test performance may not be appropriate for these new testing contexts. In particular, the notion that examinee performance is best represented by points on a unidimensional scale may not be valid. Alternative models could include multiple dimensions (e.g. Ackerman, Gierl, & Walker, 2003; Ackerman, 1994, 1996), subscores (e.g., Hamilton, Nussbaum, Kupermintz, Kerkhoven, & Snow, 1995; Nussbaum, Hamilton, & Snow, 1997), clusters or classes (e.g., Bolt, Cohen, & Wollack, 2001; Brown, 2000; Haertel, 1992; Sireci, 1995; Sireci, Robin, & Patelis, 1997), even hierarchies of cognitive ability states (e.g., Buck & Tatsuoka, 1998; Leighton, Gierl, & Hunka, 2004; Tatsuoka 1995; Tatsuoka, Corter, & Tatsuoka, 2004).

Given this range of possible test structures, methods to determine these test structures that are exploratory in nature may be helpful. This is due mainly to two factors. First, the structure of examinee performance on a test or assessment may not be known with certainty. As mentioned above, the failure to correctly identify test structure could result in errors in estimation of both test parameters and of student achievement. An exploratory method may be able to help identify candidate structures that could be subsequently evaluated for fit. Second, many of the methods mentioned above are limited in the range of possible structures that could be identified. For example, factor analytic methods and traditional methods of reliability assume that test structure is best described by the number of latent dimensions. A description of student performance based on, for example, membership in one of a small number of classes is not supported by such methods. Alternative methods that have minimal assumptions about the range of test structures and the power to detect key relationships in test performance would address these two factors. But, what type of method could be used?

A starting point in identifying potentially useful methods is to examine proven methods from other disciplines that effectively solve similar analytical problems. One class of methods that has been shown to be particularly powerful in finding dominant relationships in complex data is known as artificial neural networks (ANNs).

This thesis is an investigation of the utility of one type of ANN, the Kohonen Self-Organizing Map (SOM) in the determination of test structure (Kohonen, 1983, 1990, 2001). The SOM is a particularly promising method for this purpose as it has been shown to be able to detect and represent relationships in complex data in a more simplified form: a projected, map-like space. A characteristic of this space that could be

important in its role as a method for determining test structure is that it is *topology-preserving*. That is, the SOM represents the probability density of the original data and also produces a continuous, ordered representation of it. This characteristic may be important to educational measurement because it suggests that the dominant, relevant aspects of examinee performance may be reflected in the topology of the SOM. Therefore, when applied to educational data, the organization of the map may reveal test structure and therefore, is worth investigating.

The organization of this thesis is as follows. First, a review of exploratory methods designed to determine test structure is undertaken. Chapter 2 discusses methods according to their usefulness in determining the number of dimensions in test data. Chapter 3 focuses specifically on those methods designed to classify performances into one of a finite number of categories. Chapter 4 introduces Artificial Neural Networks, specifically focusing on the SOM. Chapters 5 through 7 detail a series of experiments designed to test the capabilities of SOMs in detecting test structure. Chapter 8 interprets the results from those experiments and discusses their implications for educational measurement and suggestions for future research. Through this research, it will be argued that with a clear understanding of the conditions that support their use, SOMs could be an important and appropriate tool for determining the test structure in certain educational measurement applications.

References

Ackerman, T. A. (1994). Using multidimensional item response theory to understand what items and tests are measuring. *Applied Measurement in Education, 7*, 255-278.

Ackerman, T. A. (1996). Graphical representation of multidimensional item response theory analyses. *Applied Psychological Measurement, 20*, 311-329.

Ackerman, T. A., Gierl, M. J., & Walker C. (2003). Using multidimensional item response theory to evaluate educational and psychological tests. *Educational Measurement: Issues and Practice, 22,* 37-53.

Bolt, D.M., Cohen, A.S., & Wollack, J.A. (2002). Item parameter estimation under conditions of test speededness: Application of a mixture Rasch model with ordinal constraints. *Journal of Educational Measurement, 39*, 331-348.

Brown, R. S. (2000, April). *Using latent class analysis to set academic performance standards*. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

Buck, G., & Tatsuoka, K. K. (1998). Application of the rule-space procedure to language testing: Examining attributes of a free response listening test. *Language Testing, 15,* 119-157.

Gessaroli, M. E., & De Champlain, A. F. (1996). Using an approximate chi-square statistic to test the number of dimensions underlying responses to a set of items. *Journal of Educational Measurement, 33,* 157-179.

Gitomer, D. H., & Yamamoto, K. (1991). Performance modeling that integrates latent trait and class theory. *Journal of Educational Measurement, 28*, 173-189.

Haertel, E. H. (1992, April). *Latent traits or latent states? The role of discrete models for ability and performance*. The Raymond B. Cattell Award Invited Address presented at the Annual Meeting of the American Educational Research Association, San Francisco, CA.

Hambleton, R. K., & Rovenelli, R. J. (1986). Assessing the dimensionality of a set of test items. *Applied Psychological Measurement, 10,* 287-302.

Hamilton, L. S., Nussbaum, E. M., Kupermintz, H., Kerkhoven, J. I. M., & Snow, R. E. (1995). Enhancing the validity and usefulness of large-scale educational assessments: II. NELS:88 science achievement. *American Educational Research Journal, 3,* 555-581.

Hattie, J.A. (1985). Methodology review: Assessing unidimensionality of tests and items. *Applied Psychological Measurement, 9,* 139-164.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43,* 59-69.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE, 78,* 1464-1480.

Kohonen, T. (2001). *Self-organizing maps* (3rd ed.). Berlin: Springer-Verlag.

Kuder, G. F., & Richardson, M. W. (1937). The theory of estimation of test reliability. *Psychometrika. 2,* 151-160.

Leighton, J. P., Gierl, M. J., & Hunka, S. (2004). The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach. *Journal of Educational Measurement, 41,* 205-236.

McDonald, R. P. (1967). *Non-linear factor analysis*. Psychometric Monographs, No. 15.

Mislevy, R.J. (1987). Recent developments in item response theory. *Review of Research in Education, 15*, 239-275.

Mislevy, R. J. (1996). Test theory reconceived. *Journal of Educational Measurement, 33*, 379-416.

Mislevy, R. J., Almond, R. G., Yan, D., & Steinberg, L. S. (2000). Bayes nets in educational assessment: Where do the numbers come from? (Technical Report no. 518). Los Angeles, California: California University, Center for the study of evaluation.

Nandakumar, R. (1994). Assessing dimensionality of a set of item responses—comparison of different approaches. *Journal of Educational Measurement, 31,* 17-35.

Nussbaum, E. M., Hamilton, L. S., & Snow, R. E. (1997). Enhancing the validity and usefulness of large-scale educational assessments: IV. NELS:88 science achievement to twelfth grade. *American Educational Research Journal, 34,* 151-173.

Rudner, L.M. & Liang, T. (2002). Automated essay scoring using Bayes' Theorem. *The Journal of and Assessment, 1,* 3-21.

Sireci, S. G., (1995, August). *Using cluster analysis to solve the problem of standard setting*. Paper presented at the annual conference of the American Psychological Association, New York, NY, August.

Sireci, S. G., Robin, F., & Patelis, T. (1999). Using cluster analysis to facilitate standard setting. *Applied Measurement in Education, 12*, 301-325.

Sireci, S. G., Thissen, D., & Wainer, H. (1991). On the reliability of testlet-based tests. *Journal of Educational Measurement, 28*, 237-247.

Stevens, R. & Palacio-Cayetano, J. (2003). Design and performance frameworks for constructing problem-solving simulations. *Cell Biology Education*, *2*, 162-179.

Stevens, R., Ikeda, J., Casillas, A., Palacio-Cayetano, J., & Clyman, S. (1999). Artificial neural network-based performance assessments. *Computers in Human Behavior, 15*, 295-313.

Stevens, R., Johnson, D., & Soller, A. (2005). Probabilities and predictions: Modeling the development of scientific problem-solving skills. *Cell Biology Education*, *4*, 42–57.

Stevens, R., Soller, A., Cooper, M., & Sprang, M. (2004). Modeling the development of problem-solving skills in chemistry with a web-based tutor. In Lester, Vicari, & Paraguaca (Eds), *Intelligent Tutoring Systems, 7th International Conference Proceedings* (pp. 580-591), Berlin, Germany: Springer-Verlag.

Stout, W. (1987). A nonparametric approach for assessing latent trait unidimensionality. *Psychometrika*, 52, 589-617.

Stout, W., Habing, B., Douglas, J., Kim, H.R., Roussos, L., & Zhang, J. (1996). Conditional covariance based nonparametric multidimensionality assessment. *Applied Psychological Measurement*, *20*, 331-354.

Tate, R. L. (2004). Implications of multidimensionality for total score and subscore performance. *Applied Measurement in Education*, *17*, 89–112.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement 20,* 345-354.

Tatsuoka, K. K. (1990). Toward an integration of item-response theory and cognitive error diagnoses. In N. Fredericksen, R. L. Glaser, A. M. Lesgold, & M. G. Shafto (Eds.), *Diagnostic Monitoring of Skill and Knowledge Acquisition*. Hillsdale, NJ: Erlbaum.

Tatsuoka, K. K. (1995). Architecture of knowledge structures and cognitive diagnosis: A statistical pattern recognition and classification approach. In P. Nichols, S. F. Chipman, & P. L. Brennan (Eds.), *Cognitively Diagnostic Assessment* (pp. 327-359), Hillsdale, NJ: Erlbaum.

Tatsuoka, K. K., Corter, J. E., & Tatsuoka, C. (2004). Patterns of diagnosed mathematical content and process skills in TIMSS-R across a sample of twenty countries. *American Educational Research Journal, 41*, 901-926.

Walker, C. M. & Beretvas S. N. (2003). Comparing multidimensional and unidimensional proficiency classifications: Multidimensional IRT as a diagnostic aid. *Journal of Educational Measurement 40*, 255-275.

Williamson, D. M. & Bejar, I. I. (2000, April). *Kohonen self-organizing maps in validity maintenance for automated scoring of constructed response*. Paper presented at the annual meeting of the National Council on Measurement in Education. New Orleans, LA.

Williamson, D. M., I. I. Bejar, & A. S. Hone. (1999). 'Mental Model' comparison of automated and human scoring. *Journal of Educational Measurement 36*, 158-84.

Williamson, D. M., I. I. Bejar, & A. Sax. (2004). Automated tools for subject matter expert evaluation of automated scoring. *Applied Measurement in Education, 17*, 323-357.

Yamamoto, K. (1989). *HYBRID model of IRT and latent class models*. (ETS research report RR-89-41). Princeton, NJ: Educational Testing Service.

Yamamoto, K. & Gitomer, D. H. (1993). Application of a HYBRID model to a test of cognitive skill representation, in N. Frederiksen, R. J. Mislevy, & I. I. Bejar, (Eds.), *Test Theory for a New Generation of Tests*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Zenisky, A. L., Hambleton, R. K., & Sireci, S. G. (2002). Effects of local item dependencies on the validity of item, test, and ability statistics. *Journal of Educational Measurement, 39*, 1-16.

Zhang, J. & Stout, W. (1999). The theoretical detect index of dimensionality and its application to approximate simple structure. *Psychometrika, 64*, 213-249.

Chapter 2 – Methods to Determine Test Structure

Overview

As introduced in Chapter 1, identifying test structure is a critical step in creating a valid interpretation of test performance. The identification of this structure is predicated on determining the relationships among performance on items and tasks and these relationships, in turn, inform the type of measure appropriate for summarizing test performance as a whole. Furthermore, it was argued that *exploratory* methods may be helpful in contemporary measurement contexts (e.g., assessments involving performance tasks) since the data derived from these contexts may be more complex than those for which traditional methods of test structure analysis were developed (e.g., Levy & Mislevy, 2004; Stevens, Johnson, & Soller, 2005; Williamson, Bejar, & Sax, 2004). There are several of these exploratory methods used in educational measurement and the present and following chapters examine prominent examples. The examples in the present chapter have in common their description of test structure in terms of continuous dimensions. These may be contrasted with a class of methods outlined in Chapter 3, those that describe test structure as comprising a finite number of ability categories or classes. An analysis of both classes of methods, the dimensional methods in the present chapter and categorical methods in the following chapter will be helpful in providing a deeper understanding as well as a balanced evaluation of the artificial neural network approach.

Dimensional-Based Methods for Determining Test Structure

Dimensional test structure methods describe examinee performance in terms of a location on a continuous scale. From the psychological perspective, these scales can be

thought of as representing hypothetical latent traits, such as ability or achievement. The amount of this trait possessed by examinees is assumed to be continuous (and often, normally distributed) throughout a population of similar examinees. The amount of the trait possessed by a given examinee is the object of measurement for the test. For example, student performance on an achievement test is often a single number equal to the total number of items answered correctly. Ideally, this number represents an individual student's possession of the trait measured by the test as a whole.

From a statistical perspective, dimensions are manifest to the extent that common variance is present across various measures. In particular, when performance on independently functioning items is found to be related to a small number of statistical factors, the factors are interpreted as explanatory variables for that performance. For example, if strong positive correlations are found between performances on items, this may be interpreted as evidence for a variable such as ability underlying performance on the items.

In this chapter, the characteristics of several exploratory dimensional accounts of test performance are reviewed, particularly from the perspective of their similarities and differences. First, characteristics common to each dimensional account are discussed followed by the review of several prominent methods designed to create such accounts. The methods reviewed are factor analysis, multidimensional scaling (MDS), and a nonparametric procedure using conditional covariances known as DETECT.

Several characteristics define dimensional accounts of examinee performance. First, these accounts involve data reduction. That is, all dimensional methods presented here use a small number of dimensions to describe data from a large multidimensional

space, usually responses to test items. The dimensions chosen are those that, from a statistical perspective, best represent the data as a whole. A high-quality dimensional account of exam performance depends upon the fit of the small number of dimensions to the data from the larger space.

A second characteristic common to dimensional methods is the determination of relatedness among observations. In many methods to determine test structure, this relatedness is derived from correlations or covariances between items. This is because it is assumed that essential information about test dimensionality is contained in the correlation or covariance among items (e.g., McDonald, 1982). Other means to determine relatedness include those applied at the level of examinee and not item performance. For example, a matrix of similarity between different examinee's responses could serve as the basis for determining test structure. In the present chapter, item relatedness is the basis of similarity for factor analysis and DETECT, while examinee relatedness is used for the application of MDS.

Third, dimensional methods lead to the identification of dominant or principal directions of variation in the matrices described above. The means by which various methods accomplish this vary considerably. For example, factor analysis can proceed by applying a variety of statistical techniques to the principal directions problem, such as principal components analysis (PCA), or maximum likelihood estimation (MLE). Multidimensional scaling typically uses an iterative approach to determine dimensional co-ordinates of individual observations that best preserve between-observation distances, and the DETECT procedure uses a genetic algorithm to help determine the optimal partition of items into dimensionally homogeneous sets.

The last characteristic of these methods is that they typically involve an evaluation of the fit of the low-dimensional representation produced by the method to the data as a whole. In factor analysis, this can be accomplished by examining the variance accounted for by all factors, while in MDS the value of a stress index (e.g., Kruskal, 1974) can be examined. The DETECT procedure also features its own index of multidimensionality based on the structure of conditional covariances between item pairs (Stout, Habing, Douglas, & Kim, 1996). Each of these characteristics is examined below in the context of their respective methods.

Factor Analysis

Factor Analysis is a method used to account for variance in a set of observed variables in terms of underlying hypothetical or latent factors (unobserved explanatory variables). For example, in educational measurement, it is important to know whether students' scores on math test items (observed variables) can be explained in terms of more basic variables such as students' overall mathematics and spatial ability (latent factors). These more basic variables are the dimensions upon which student ability or achievement may vary and therefore can form the basis of a summary of test performance.

To see this more clearly, consider a test that is designed to be unidimensional, that is, a single latent dimension is assumed to account for differences in examinee test performance. Factor analysis could be applied to correlations among examinees' item responses to determine if this assumption is correct and thus only a single dimension is identified. Suppose that after factor analysis, it was discovered that two independent factors account for variance in the responses. From this result, the test developer could

determine that the use of a unidimensional scoring model such as IRT may be inappropriate and that a multidimensional version would be preferred.

How Does Factor Analysis Work?

Conceptually, factor analysis proceeds by partitioning the variance from each observed variable (e.g., a test item) into that which is shared with other observed variables (e.g., remaining test items) and variance unique to itself. When variance is shared across many items, it normally indicates the presence of an explanatory factor. The factor solution containing the fewest number of interpretable factors accounting for the greatest amount of variance is typically adopted as the factor analytic solution (e.g., Kim & Mueller, 1978; McDonald, 1985).

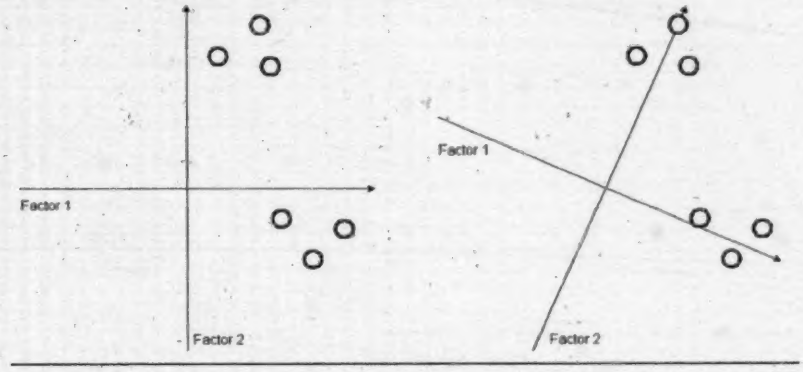
More technically, factor analysis begins with a correlation or covariance matrix of the observed variables. Then, one of several statistical procedures (e.g., maximum likelihood estimation, principal components analysis) is employed to create candidate factor models that are able to account for the correlations or covariances with a minimum of error. These models specify both the number of common factors needed and weights or coefficients that quantify the extent to which each of the observed variables is dependent on the factor, as indicated in the following formula:

$$\boldsymbol{R = FPF' + U} \qquad (2.1)$$

where $\boldsymbol{R}$ is the correlation matrix of observed variables, $\boldsymbol{F}$ is a matrix that represents the weighting of each observed variable in the definition of each common factor (e.g. eigenvectors), $\boldsymbol{P}$ is a diagonal matrix that represents the dominance of each factor in the

data (e.g., eigenvalues), and ***U*** is a matrix of the unique variance in each observed variable. The weights in ***F*** can be used to plot each observed variable on a space whose co-ordinate axes are defined by the factors in the solution. Further, to aid in the interpretability of the factor solution, axes can be *rotated* to improve the interpretability of the factor structure (see Figure 2.1).

Figure 2.1. Unrotated versus rotated factor solutions.



Determining test structure in terms of the number of underlying dimensions is an important application of factor analysis in education measurement (e.g., Hambleton & Rovenelli, 1986; Nandakumar, 1994; Gessaroli & De Champlain, 1996). Correctly identifying dimensionality is of particular importance for IRT models where the accuracy of parameter estimates (i.e., item difficulty, discrimination, and examinee ability) depends upon the local independence of items. If all the dimensions of examinee ability that predict performance on the test are not accounted for, the local independence assumption may be violated. This could result in inaccurate parameter estimates. Since IRT has been shown to be a variant of non-linear factor analysis (NLFA, e.g., McDonald,

1982; Takane & deLeeuw, 1987; Knol & Berger, 1991), NLFA is seen to be particularly appropriate for identifying underlying factors (e.g., Hattie, 1985; Hattie, Krakowski, Rogers, & Swaminathan, 1996; Gessaroli & De Champlain, 1996).

How sensitive is factor analysis in detecting the dimensionality of data? Gessaroli and De Champlain (1996) address this question by comparing a χ^2 statistic based on NLFA with Stout's (1987) T statistic (DIMTEST). The NLFA/ χ^2 approach differs from DIMTEST in that the former tests the null hypothesis that the conditional correlations between item pairs is zero after variance due to identified factors is removed while the latter compares the average variance accounted for by total test score between 2 subsets of items: those thought to be representative of the test as a whole with those thought most likely to be multidimensional. Though NLFA had been favourably compared with various methods of dimensionality assessment (e.g., linear factor analysis, Mislevy, 1986; Hambleton & Rovenelli, 1986; DIMTEST, Nandakumar, 1994; Hattie, Krakowski, Rogers, & Swaminathan, 1996; Mantel-Haenszel, Nandakumar, 1994), no clear criteria had been established for the sufficiency of a given factor model (i.e., when a given factor accounts for enough variance to be included in the factor model). Thus, Gessaroli and deChamplain (1996) conducted a simulation study to evaluate the χ^2 statistic based on NLFA as a criterion for determining this sufficiency and also to compare its sensitivity with Stout's T.

The simulated data sets in this study varied along several dimensions: dimensionality (1, 2), sample size (500, 1000), number of items (15, 30, 45), test reliability ("weak", "moderate", "strong"), and for 2 dimensional tests, the proportion of items loading on each dimension (50:50, 80:20). One hundred replications were

performed for each of the above 54 conditions. In general, the performance of the χ^2 statistic and thus the sensitivity of NLFA to dimensionality was at least as good as, and in some cases, better than Stout's (1987) T. In particular, the performance of the χ^2 statistic was markedly better for the 2 dimensional data sets that had 15 items, weak reliability, and an unequal proportion of test items reflecting each dimension.

The study by Gessaroli and De Champlain (1996) shows that factor analysis is an effective tool in the determination of the dimensionality of responses to test items. Given the close relationship between the non-linear version of factor analysis employed in this study and IRT, this result shows that NLFA is a valuable tool for determining the appropriateness of ability parameters from a unidimensional IRT model as a summary for test performance.

Several caveats about the use of factor analysis as a method to determine test structure are worth noting. First, the determination of the number of factors in the solution is somewhat arbitrary. Gessaroli and De Champlain (1996) highlight this limitation, as the lack of definitive criteria in determining dimensionality motivated their research. Second, the interpretation of a factor analysis solution can be ambiguous. A particular version of this problem is the interpretation of a rotated solution. Two different rotations can yield different, but still plausible interpretations of the factor structure. A classic example of this ambiguity is Thurstone's (1934) reinterpretation of Spearman's (1904) *g*, or single general intelligence factor based on a rotation of co-ordinate axes. From this rotation Thurstone argued for the multi-faceted nature of intelligence and thus set the stage for a debate that continues to the present day.

In summary, factor analysis is a technique of statistical data reduction that accounts for the variance in a data set with a large number of observed variables in terms of a much smaller number of common factors. For the purpose of determining test structure, a factor analysis can identify the number of underlying dimensions on which examinees' performance differ, a critical step in determining an appropriate summary of performance. However, the number of dimensions identified by factor analysis is not always unequivocal, as are possible interpretations of the factor structure. As shall be seen in the section to follow, multidimensional scaling addresses some of these limitations and thus offers an alternative to factor analytic models in creating a dimensional account of test structure.

Multidimensional Scaling

Multidimensional scaling (MDS) is a statistical method that uses "distance-like" measures between observations in a data set to create a lower-order spatial (i.e., map-like) representation of its essential structure (e.g. Davison, 1983; Kruskal & Wish, 1978). Thus, MDS, like factor analysis, is a method of data reduction.

A key advantage of MDS for educational measurement is the interpretability of its solution. That is, it can reveal variables or dimensions (e.g., requisite abilities, content areas, item difficulty) that influence student's test performance and display them in a highly interpretable spatial format. How does MDS accomplish this?

Take, for example, a geography test comprising 50 items. Before analysis, each item could be viewed as a separate dimension on which examinees are scored. However, when these items are analyzed using MDS, suppose it is found that two dimensions are sufficient to represent a large proportion of variance from the original 50 items. Given

the strength of these dimensions, the map-like representation that MDS provides will likely be straightforward to interpret by inspection. For instance, easy and hard geography items could be located at either end of the first dimension, while the second dimension could differentiate items involving reasoning with maps from those not involving maps. As an educational researcher, this information could be used as evidence for the importance of knowledge and skill in geography as a key factor in test performance, as well as the perhaps unexpected importance of item format. This may lead to the adoption of a scoring scheme that reflects the importance of each component: knowledge of geography and facility with maps.

How Does MDS Work?

MDS works by deriving a mathematical function that relates distances between observations from the original high dimensional space to distances between those same observations in a space with much smaller dimensionality (Kruskal & Wish, 1978). The goal of this function (called the *Stress* function) is to minimize the difference of the distances between same observations in the two spaces. This function is defined as follows,

$$Stress = \sqrt{\frac{\sum_i \sum_j (d_{ij} - \delta_{ij})^2}{\sum_i \sum_j \delta_{ij}^2}} \qquad (2.2)$$

where d_{ij} is the distance between pairs of observations i and j in the original multidimensional space and δ_{ij} is the distance between those same observations in the new space created by MDS. It is possible for MDS to achieve very good match between

distances in the two spaces when there are common dimensions that underlie performance on many of the observations. When this is the case, the common dimensions will likely form the axes of the MDS solution.

MDS has been used extensively in the educational measurement literature. Prominent applications include determining the dimensional structure of items (e.g., Bolt, 2001; deAyala & Hertzog, 1991; Meara, Robin, & Sireci, 2000; Oltman, Stricker, & Barrows, 1990; Sireci & Khaliq, 2002) and providing evidence of content and construct validity (e.g., Deville, 1996; Sireci & Geisinger, 1992, 1995; Sireci, 1998).

The study by DeAyala and Hertzog (1991) provides an illustrative example of how MDS can be used to determine dimensional structure. In this study, MDS was used to recover the number of dimensions in simulated test data, data that varied with respect to the number of underlying dimensions (1, 2), the proportion of items tapping into each dimension (50/50, 64/36), and the correlations among those dimensions (0.01, 0.10, 0.60). The performance of MDS in recovering the dimensionality of the data relative to a factor analysis of tetrachoric correlations was the specific focus of this research.

The data serving as input to MDS were various distance measures between vectors of scored responses to items. Five measures were investigated, Euclidian distance, squared Euclidian distance, cosine, block distance, and Chebychev distance. For each analysis, the value for the stress index was examined relative to Kruskal and Wish's (1978) criteria[1] and with respect to the prominence of the 'elbow' in stress plots. The factor analysis solutions were evaluated by examining the percentage of variance accounted for by each factor, a chi-square measure determining the fit of each factor

[1] The criteria states that an elbow above a stress value of 0.10 should only be accepted for the one-dimensional solution, and only if it occurs at stress less than 0.15.

solution, the number of factors for which eigenvalues were greater than one, and the prominence of the elbow in scree plots of the eigenvalues.

DeAyala and Hertzog concluded that factor analysis produced equivocal results for both the one-dimensional data and for two-dimensional where the dimensions were highly correlated. In particular, the one-dimensional data were not unanimously identified as such by all methods, and the second factor in the highly correlated two-dimensional data was not clearly identified by factor analysis. In contrast, when MDS was used with Euclidian and squared Euclidian distances, the dimensionality of all data sets was correctly identified. As a result, DeAyala and Hertzog concluded that MDS using these distance measures could play an important role in determining the appropriateness of a unidimensional versus multidimensional IRT model for a given set of data.

The above example highlights several essential characteristics of MDS analysis, particularly in comparison to factor analysis. Similar to factor analysis, MDS creates a dimensional account of test structure. In factor analysis the primary task is to extract those dimensions from correlations among items, while the primary task in MDS is to preserve distance relationships between vectors of examinee's responses to test items. Last, MDS provides a map-like representation of the examinee data which was subsequently interpreted in terms of the number of dimensions. However, as with factor analysis, determining the precise number of dimensions in the data is somewhat arbitrary as it depends upon the specific criteria employed.

DETECT

The DETECT method (Kim, 1994, Zhang & Stout, 1999) produces an index of the dimensionality of test data based on the structure of conditional covariances between item pairs. In particular, the procedure looks for clusters of items whose item pairs have similar covariances after variance due to the total test score has been factored out. After the test has been optimally partitioned into multidimensional item clusters, test developers can determine the overall multidimensionality of the test, the extent to which dimensions on the test approximate simple structure, and the specific items that correspond to each unique dimension. Knowing which items belong together in clusters can lead to understanding about the substantive nature of the dimensional structure and inform decisions regarding future test design and scoring summaries.

Take for example, a test containing several passages of text that students are to read and upon which they answer a number of questions. The DETECT procedure could determine whether the passages induced multidimensionality in student responses by first, factoring out the variance due to the total score, then examining the remaining inter-item covariance matrix. Suppose that clusters of items containing only those relating to specific passages had the most similar covariances, while items from separate passages had covariances that were very different. Since the items that defined the clusters all relate to certain passages, the test developer would have evidence that the existence of sets of items all based on the same passages leads to multidimensionality in the test. The value of the DETECT index, D_{Max}, would help the developer determine whether the extent of the multidimensionality was of concern and adjustments needed to be made in terms of the scoring scheme or test design.

How does DETECT work?

The DETECT procedure comprises three main steps. First, test variance due to the dominant dimension is factored out. Second, covariances for all item pairs are calculated. Third, an optimal partition of the test into item clusters is made using a combination of hierarchical cluster analysis (HCA) and a genetic algorithm. Two indexes can then be calculated; D_{Max} reveals the strength of multidimensionality present on the test, while r_{Max} determines the extent to which the test approximates simple structure.

The first step in the DETECT method is accomplished by separating test scores into n_i cells, where each cell contains responses from examinees who answered the same number of items correctly. Then, an average covariance for each item pair i, j across all sum score levels is calculated and weighted by the number of students achieving each score point. This value is known as the *conditional covariance* of item pair i, j. The third step requires a partition of items into clusters based on the similarity of their covariances. As a result of the process that partials out variance due to total test score, covariances of item pairs that belong to the same cluster are positive, while those belonging to different clusters are often negative. A consequence of these differences in sign is that when an optimal partitioning of the test is achieved, covariances of items from different clusters *subtracted* from covariances of items in the same clusters will reach a maximum value for that test. Since determining the optimal partition for the test can involve evaluating an inordinately large number of possible partitions, Zhang and Stout used an optimization technique well proven in other domains: a genetic algorithm. The algorithm begins with a provisional partition of items into cluster based on HCA, then "mutates" the clusters by changing the cluster membership of several items. Solutions based on these mutations

that lead to higher values of the DETECT index are kept, and characteristics of these successful mutations are used as the basis of subsequent mutations. The procedure is complete when no further improvements in the D_{Max} index are observed. To determine the amount of multidimensionality present, the value of D_{Max} is typically compared against benchmarks produced by Stout et al. (1996).[2]

Zhang and Stout (1999) tested the efficacy of the DETECT procedure on items from the Analytical Reasoning section of the GRE exams. They used data from 2477 examinees on 19 items that corresponded to four different passages on this test. Using the DETECT procedure, they found that the optimal partition of the test into homogeneous clusters that revealed the greatest amount of multidimensionality was that which clustered items according to passages. Furthermore, the value of the DETECT index for this partitioning was very large, indicating a significant amount of multidimensionality present.

In summary, the DETECT method represents an innovative approach to determining the structure of test data. It relies upon the expected nature of the covariance structure of data once the overall direction of best measurement has been accounted for. Then, a search for the optimal partition relies upon a method imported from other computational and information processing domains: the genetic algorithm. Its success in the DETECT context underscores the potential value of incorporating methods from such domains to problems in educational measurement. It is perhaps notable that the present research applying Artificial Neural Networks to determining test structure represents an approach similar in kind.

---

[2] More recently, Gierl, Leighton, & Tan (in press) demonstrate that approximation to simple structure is a prerequisite for correctly interpreting the DETECT index and consequently offer additional guidelines in terms of both D_{Max} and r_{Max}.

Summary

All of the above methods use statistical relationships in test data to identify test structure in terms of a small number of continuous dimensions. The relationships capitalized upon vary from method to method: variance extracted from correlation matrix; fitting of inter-item "distances" to a smaller dimensional space; identification of clusters of homogeneous items based on conditional covariance. These methods provide critical information in terms of the latent structure of responses and test items and therefore can provide support for particular types of dimensionally-based scoring models. An important question in educational measurement is whether the latent structure of the test is most closely related to dimensional structure or whether there are other candidate models that can better account for the latent structure of the test.

In the next chapter, latent class models as methods to determine test structure are discussed. These models are important because they offer alternatives to dimensionally-based models and therefore provide test designers with different models of test performance.

References

Bolt, D.M. (2001). Conditional covariance-based representation of multidimensional test structure. *Applied Psychological Measurement, 25*, 244-257.

Davison, M.L. (1983): *Multidimensional Scaling*. John Wiley and Sons, New York.

DeAyala, R. J. & Hertzog, M. A. (1991). The assessment of unidimensionality for use in item response theory. *Multivariate Behavioral Research, 26,* 765-792.

Deville, C. W. (1996). An empirical link of content and constraint evidence. *Applied Psychological Measurement, 20*, 127-139.

Gessaroli, M. E., & De Champlain, A. F. (1996). Using an approximate chi-square statistic to test the number of dimensions underlying responses to a set of items. *Journal of Educational Measurement, 33,* 157-179.

Gierl, M. J., Leighton, J. P., & Tan, X. (in press). Evaluating DETECT classification accuracy and consistency when data display complex structure. *Journal of Educational Measurement*.

Hambleton, R. K., & Rovenelli, R. J. (1986). Assessing the dimensionality of a set of test items. *Applied Psychological Measurement, 10,* 287-302.

Hattie, J.A. (1985). Methodology review: Assessing unidimensionality of tests and items. *Applied Psychological Measurement, 9*, 139-164.

Hattie, J., Krakowski, K., Rogers, H. J., & Swaminathan, H. (1996). An assessment of Stout's index of essential unidimensionality. *Applied Psychological Measurement, 20,* 1-14.

Kim, H. R. (1994). *New techniques for the dimensionality assessment of standardized test data*. Unpublished Doctoral Dissertation. University of Illinois at Urbana-Champaign, Department of Statistics.

Kim, J.O. & Mueller, C.W. (1978). *Factor analysis: statistical methods and practical issues*. Thousand Oaks, CA: Sage Publications.

Knol, D. L., & Berger, M. P. F. (1991). Empirical comparison between factor analysis and multidimensional item response models. *Multivariate Behavioral Research, 26*, 457-477.

Hattie, J., Krakowski, K., Rogers, H. J., & Swaminathan, H. (1996). An assessment of Stout's index of essential unidimensionality. *Applied Psychological Measurement, 20*, 1–14.

Kruskal, J. B. (1964). Nonmetric multidimensional scaling: A numerical method. *Psychometrika, 29*, 115-129.

Kruskal, J. B., & Wish. M. (1978). *Multidimensional Scaling*. Beverly Hills, CA: Sage Publications.

Levy, R., & Mislevy, R. J (2004). Specifying and refining a measurement model for a computer-based interactive assessment. *International Journal of Testing, 4*, 333-369.

McDonald, R. P. (1982). Linear versus non-linear models in item response theory. *Applied Psychological Measurement, 6*, 379-396.

McDonald, R. P. (1985). *Factor analysis and related methods*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Meara, K. P., Robin, F., & Sireci, S. G. (2000). Using multidimensional scaling to assess the dimensionality of dichotomous item data. *Multivariate Behavioral Research, 35,* 229-259.

Mislevy, R.J. (1986). Recent developments in the factor analysis of categorical data. *Journal of Educational Statistics, 11,* 3-31.

Nandakumar, R. (1994). Assessing dimensionality of a set of item responses - Comparison of different approaches. *Journal of Educational Measurement, 31,* 17-35.

Oltman, P. K., Stricker, L. J., & Barrows, T. S. (1990). Analyzing test structure by multidimensional scaling. *Journal of Applied Psychology, 75,* 21-27.

Sireci, S. G. (1998). Gathering and analyzing content validity data. *Educational Assessment, 5,* 299-321.

Sireci, S. G. & Geisinger, K. F. (1992). Analyzing test content using cluster analysis and multidimensional scaling. *Applied Psychological Measurement, 16,* 17-31.

Sireci, S. G., & Geisinger K. F. (1995). Using subject matter experts to assess content representation: An MDS analysis. *Applied Psychological Measurement, 19,* 241-255.

Sireci, S. G., & Khaliq, S. N. (2002). NCME members' suggestions for recruiting new measurement professionals. *Educational Measurement: Issues and Practice, 21,* 19-24.

Spearman, C. (1904). "General intelligence," Objectively determined and measured. *American Journal of Psychology, 15,* 201-293.

Stevens, R., Johnson, D., & Soller, A. (2005). *Cell Biology Education, 4,* 42–57.

Stout, W. (1987). A nonparametric approach for assessing latent trait unidimensionality. *Psychometrika, 52*, 589-617.

Stout, W., Habing, B., Douglas, J., Kim, H.R., Roussos, L., & Zhang, J. (1996). Conditional covariance based nonparametric multidimensionality assessment. *Applied Psychological* Measurement, *20*, 331-354.

Takane, Y. & DeLeeuw, J. (1987). On the relationship between item response theory and factor analysis of discretized variables. *Psychometrika, 52*, 393-408.

Thurstone, L. L. (1934). The vectors of mind. *Psychological Review, 41*, 1-32.

Williamson, D. M., I. I. Bejar, & A. Sax. (2004). Automated tools for subject matter expert evaluation of automated scoring. *Applied Measurement in Education, 17*, 323-357.

Zhang, J. & Stout, W. (1999). The theoretical DETECT index of dimensionality and its application to approximate simple structure. *Psychometrika, 64*, 213-249.

Chapter 3 – Latent Class Accounts of Test Structure

In Chapter 2, methodologies describing test structure in terms of underlying continuous dimensions were reviewed. In Chapter 3, the review of these methodologies continues, but changes focus to consider those that describe test structure in terms of latent states. These models imply that performance is best described by examinee membership in one of a small number of classes, rather than the possession of an 'amount' of a latent trait implied by dimensional methods. Important for educational measurement, membership in a specific class could represent the possession of a particular set of skills and knowledge underlying test performance. For example, on a test involving subtraction of mixed fractions (Mislevy, Yan, Almond, & Steinberg, 2000), one class could represent those students that are able to subtract fractions with the same denominator and separate the whole number from the fraction, but are unable to simplify a mixed number. If the identified classes have clear interpretations in terms of achievement, latent class methods could identify important states of mastery on the test.

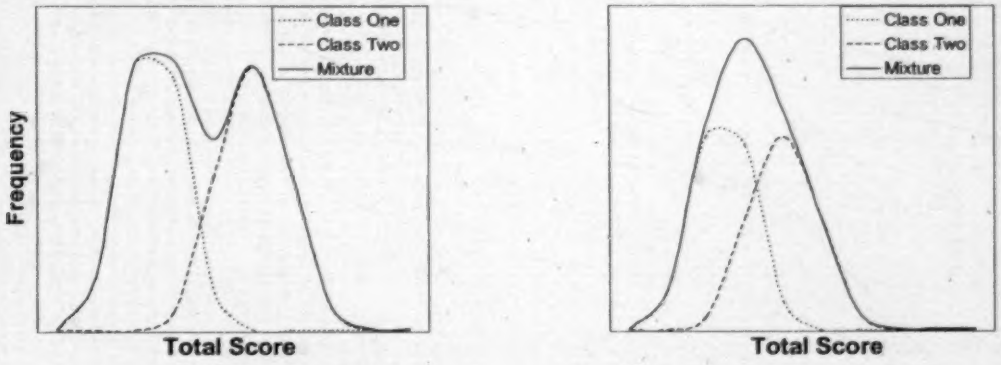
From a statistical perspective, the existence of latent states is equivalent to the existence of conditional distributions in the test data. That is, a group of examinees belonging to the same latent class will have propensities to respond to test items in similar ways, and this similarity in responses will manifest itself as a distribution in the test data. The entire test data is then simply the composite of each *class conditional* distribution, with each distribution represented by the number of examinees belonging to it. To take the mixed fraction example from above, when a student possesses all skills but the ability to reduce fractions, he should be able to perform well on items that do not demand this missing skill, and poorly on those that do. If other examinees have a similar

set of skills, the expectation would be that they would perform similarly to each other across all items on the test. This similarity in performance, in turn, implies the existence of a distribution comprising these examinees' responses.

As mentioned above, latent classes could manifest in test data as a mixture of individual class distributions (Everitt & Hand, 1981). From this conception, it would follow that regions of high probability density would emerge at the centres of the distributions, provided that different class distributions did not overlap too much. In the left-hand panel of Figure 3.1 two adjacent distributions, classes one and two are represented, as well as the probability distribution resulting from their overlap. In this case, it can be seen that the highest regions of probability density in the mixture distribution correspond to the centres of the distributions of each class. In contrast, the right-hand panel shows a mixture distribution composed of two class distributions that overlap considerably. In this case, the centre of the mixture distribution is somewhere between the two class distributions. In this case, probability density alone may be insufficient to detect the two underlying distributions. However, other information, such as the form of the underlying class distributions, could help identify the test structure.

The two test structure methods reviewed in this chapter differ in their use of probability density as a marker for the existence of a latent class. The first method reviewed, cluster analysis, relies on probability density to define a cluster structure for the data. The second method, latent class analysis, assumes a parametric form of each class distribution and therefore tries to model the shape of the composite distribution by specifying parameters of the distributions that compose it. The use of these two methods and the implications of their use for correctly identifying test structure, are reviewed next.

Figure 3.1. The Shape of Mixture Distributions According to Degree of Overlap of the Underlying Class Distributions



Class-Based Accounts for Determining Test Structure

Cluster Analysis

Cluster analysis assigns individual observations into categories based on their geometric similarity to other observations in the data set. In general, the goal of cluster analysis is to identify clusters for which each observation is most similar to other observations in the same cluster and most different from those in other clusters. A test structure definition based on cluster analysis therefore is composed of homogeneous classes, each with unique characteristics. A test structure definition derived from cluster analysis would therefore differ from those derived from dimensional procedures in that rather than assigning observations to a scale location indexed by continuous dimensions, each observation is assigned to exactly one cluster in an all-or-none manner.

Consider the following example. Suppose that a prominent model of learning explains the development of a particular math skill in terms of five discrete stages of increasing mastery. Also, suppose that the attainment of a higher stage depends upon the attainment of each previous stage. According to this model the primary predictor of

performance is the student's stage in the achievement trajectory and hence, there is good reason to expect that the data from students learning this skill would be both categorical and hierarchical. Now imagine that a cluster analysis was performed on a test comprising 40 items equally representing each stage that was administered to a group of students at various stages in the trajectory. If it was observed that the best-fitting solution from cluster analysis comprised five clusters, this could be interpreted as evidence that students are performing as predicted by the model. Furthermore, an analysis of the items answered correctly and incorrectly within each cluster in relation to the stage for which the item was designed, would provide further evidence for the correctness of predictions derived from the model. Last, a test administrator could use the cluster analysis solution to assign individual students to steps of mastery in the skill.

There has been a considerable amount of research involving the application of cluster analysis to different areas in educational measurement. These areas include the classification of items to dimensions leading to an assessment of test dimensionality (e.g., Roussos, Stout, & Marden, 1998; Tay-lim & Stone, 2000; Stout, Habing, Douglas, Kim, Roussos, & Zhang, 1996), classification of items on the basis of content (e.g., Beller, 1990; Corter, 1995) and defining categories of achievement based on test performance (e.g., Sireci, 1995; Sireci, Robin, & Patelis, 1999).

Fundamentally, cluster analysis works by grouping individual observations into classes (i.e., clusters) on the basis of similarity. Since regions of high probability density are those in which responses of different examinees are similar, it is more likely that cluster centres will be located in such regions. The precise method by which the grouping is accomplished differs between the different cluster analysis variants. The

grouping methods of the two most common variants of cluster analysis, hierarchical agglomerative and *K*-means, are cases-in-point.

In hierarchical agglomerative cluster analysis (HCA) each individual observation is initially designated as a separate cluster. In a stepwise fashion, the most similar clusters are combined into larger units, ending when there exists one super-cluster containing all observations. The solution at each step can then be evaluated for its respective fit to the data typically using a chi-squared procedure. In contrast, *K*-means cluster analysis starts with the user identifying the number of clusters desired in the solution, and the centroids (cluster means) for each. Each individual observation is compared with the values of each centroid and assigned to the cluster with which it is most similar. The values of the centroids are recalculated after comparison with all observations. The process is complete when, after a complete pass through the dataset, no re-assignments are made. Like HCA, candidate solutions comprising different numbers of clusters can be compared on the basis of their fit to the observed data.

One significant study involving cluster analysis was the recent work by Sireci (1995) to inform the setting of cut scores for standard setting. Though the study was not directly focused on determining test structure, the motivation for using cluster analysis was to determine whether the data from the writing skills assessment from the tests of General Educational Development (GED) naturally formed clusters which could be used as the basis for identification of different levels of achievement. Therefore, test structure information would form the basis of the classification into these levels.

Sireci used *K*-means cluster analysis to cluster student performances on 4 subscales of the GED, separately for two distinct administrations of the test. Three of the subscales

measured different components of knowledge about writing, while the fourth was the evaluation of a writing sample. Candidate cluster structures comprising 2 through 8 clusters were evaluated by examining correlations between student's pass/fail status implied by the cluster analysis to actual pass/fail status based on coursework. Examining in this way data from two consecutive years, it was determined that the best-fitting solution involved 5 clusters. In this solution, three of the five clusters represented students that performed significantly below average, average, and significantly above average on each of the four subscales. Notably, this set of clusters could be seen as consistent with a unidimensional structure for the test as a whole. The remaining two clusters represented deviations from a unidimensional structure, either average performance on the writing component and below average performance on the remainder of the test or average performance on the writing component and above average performance on the remainder. These clusters therefore represented selective proficiency at the subscale level.

Two observations from Sireci (1995) are central to understanding cluster analysis as a potential method to determine test structure. First, an analysis of the cluster structure was made in terms of a pattern of proficiency across each of the subscales and not in the possession of an amount of a latent trait. Examinees belonging to each cluster could therefore be characterized as possessing certain competencies for each element of the test. For example, one group of students comprising a cluster could be described as having average competency on the writing sample, but below average competency on each of the other subscales. The second observation is that the relationship between the clusters can be used to make judgments about the test structure overall. That is, examinees who

performed well in one subtest tended to perform well in others. The exception was performance on the writing section for which performance did not predict performance very well on other subscales. Therefore, it could be argued that the test has at least two abilities that underlie performance, overall writing ability (comprising knowledge about writing and writing skill) and selective skill in writing. These two observations will be crucial to understanding how neural networks will be interpreted as a possible method for determining test structure.

Several limitations are important to note regarding the use of cluster analysis as a method to determine test structure. One central problem is the determination of the number of clusters best describing the data. Both HCA and *K*-means cluster analysis can produce many solutions for a given problem but neither procedure provides information about the quality of the solutions that they identify. Typically, some criteria must be available to provide selective support for some cluster solutions over others. There are many procedures to choose from (see e.g., Milligan, 1981) and many of them select a solution that maximizes the between cluster distance while minimizing within cluster distances. Furthermore, cluster analysis will never fail to find a solution for a given dataset, even when no natural categories exist. Thus, other analyses (e.g., graphical, correlational) must be undertaken to determine whether a categorical interpretation is appropriate for the data. Last, cluster analysis uses probability density to identify clusters in data. As shown above, when individual class distributions are close together, probability density alone may not be sensitive enough to detect them. The method reviewed next, latent class analysis, attempts to overcome this limitation by making

parametric assumptions about the underlying distributions, then deriving values for those parameters.

Latent Class Analysis

Latent Class Analysis (LCA) is a statistical and computational procedure that describes data in terms of underlying discrete latent classes. Like cluster analysis, this description contrasts with that derived from latent trait approaches such as factor analysis in which data are described in terms of their relationship to an underlying continuous latent trait or traits. In essence, LCA answers the question, "What is the most likely class structure that could have given rise to the observed data?" Since the number of classes is typically much smaller than the number of items, LCA could be considered a form of data reduction (e.g., Clogg, 1995).

As an example, imagine a social studies teacher that analyzes student responses to a recent unit test on China with latent class analysis. Furthermore, imagine that the unit test was constructed to incorporate two primary topics, geography and culture. Suppose that after analysis, the best fitting model consisted of two latent classes that upon inspection correspond to the primary topics. That is, one class identified students who did well on geography but not in culture, and the other identified the reverse pattern. The teacher could use the analysis to provide evidence that her test did tap into the skills she intended and also to provide remedial exercises to students to correct their specific weaknesses.

How does LCA work?

As in other latent variable methodologies (e.g., factor analysis) two types of variables are assumed in LCA, (a) manifest or observed variables and, (b) latent or

hypothetical variables. LCA proceeds by deriving parameters of the latent variables given the values of the parameters for the observed variables. In the most common parameterization of LCA, the manifest variables are the empirical probabilities of a specific response pattern across a set of indicator variables (see Table 3.2). These indicator variables are nominal or ordinal variables, either dichotomously or polytomously scored. The corresponding latent variables are the class conditional probabilities of each score category for each item, plus the overall probabilities of belonging in each of the latent classes (see Table 3.3). In an early example of LCA, the values of the latent parameters were derived in closed form using systems of equations (e.g., Lazarsfeld & Henry, 1968, chaps. 2 and 3), assuming the true values of the observed parameters were known. Since this assumption rarely holds true and because solving systems of equations with many variables is often intractable, later approaches to LCA involved maximum likelihood estimation (MLE) of the latent variables (Goodman, 1974, 1979). The method that Goodman developed is in common use today.

Types of Models in LCA

LCA provides maximum likelihood estimates for parameters of latent class models given the observed values on indicator variables, such as test items. From where do these LC models originate and what do they comprise? Like K-means cluster analysis, the user must specify the number of classes in a hypothetical latent class model. This specification can be made either from an exploratory or from a confirmatory perspective, or more precisely, on a continuum between exploratory and confirmatory. In the exploratory mode, the user can specify a range of candidate models with a varying number of classes. Each of the models can then be assessed for its fit of the data,

typically using chi-square (χ^2) and likelihood ratio chi-square (L^2) criteria. Certain restrictions other than just the number of classes can be made on the models. For example, our social studies teacher might have the prior belief that the number of students mastering the geography items is greater than the number of students mastered culture items. Limiting the values of parameters in the model to be estimated operationalize this constraint. By providing these constraints, specific hypotheses about the class structure in the data can be tested.

Table 3.2. Manifest Joint Probabilities for 3 Dichotomous Items

Response Pattern	Manifest Probability
111	0.220
110	0.160
101	0.060
100	0.160
011	0.060
010	0.060
001	0.060
000	0.220
Total	1.000

Note. From "Latent Structure Analysis," by P. F. Lazarsfeld and N. W. Henry, 1968, p.37. Copyright 1968 by Houghton Mifflin.

Table 3.3. Latent Parameters Leading to Manifest Joint Probabilities in Table 3.2.

		Class Conditional Item Probabilities		
Class	Class Probability	1	2	3
1	0.5	0.8	0.9	0.6
2	0.5	0.4	0.1	0.2

Note. From "Latent Structure Analysis," by P. F. Lazarsfeld and N. W. Henry, 1968, p.36. Copyright 1968 by Houghton Mifflin.

Determining Test Structure

As mentioned above, the fit of each hypothesized latent class model to the data is assessed through the χ^2 test. For the case where there are 4 indicator variables, i, j, k, and l, the formula test is:

$$\chi^2 = \sum (f_{ijkl} - \hat{F}_{ijkl})^2 / \hat{F}_{ijkl} \qquad (3.1)$$

where f_{ijkl} are observed values for the highest order joint probability. $\hat{F}_{ijkl}$ are the same joint probabilities calculated using the maximum likelihood estimates. When the chi-squared value at the appropriate degrees of freedom is significant, this indicates that there is a statistically significant difference between the observed and estimated values, and thus that the latent class model is not fitting the data well. Often in LCA, the critical question is not if a single model does or does not fit the observed data but which of

several models provides the optimal fit. In this case, the L^2 likelihood ratio test [formula (3.2)] is often used.

$$L^2 = 2\sum \hat{F}_{ijkl} \ln(\hat{F}_{ijkl} / f_{ijkl}) \qquad (3.2)$$

The advantage of this statistic is that it can be partitioned (e.g., McCutcheon, 1987) and therefore allows the comparison of fit between different models. For example, one could compare the fit of two 3 classes models to a data set, one unrestricted (i.e., the values of latent parameters were not constrained) and one restricted (i.e., the values of particular parameters constrained to particular values). If the restricted model offered similar fit to the unrestricted model, it may be accepted as the most appropriate since it required the estimation of fewer parameters.

Why LCA works

In the most basic form of LCA, the assumption of conditional independence enables the derivation of unobserved parameters of latent classes from observed probabilities. (e.g., Lazarsfeld & Henry, 1968; Goodman, 1974; McCutcheon, 1987, Clogg, 1995). The assumption states that within a latent class, there is no statistical relationship between responses to different items. Under these conditions, the joint probability of correctly answering multiple items will equal the product of each individual probability, expressed as:

$$\pi_X(t) \cdot \prod_1^J \pi_J \mid X_T, \qquad (3.3)$$

where $\pi_X(t)$ is the probability that a randomly chosen examinee belongs to the class t, and $\pi_J | X_T$ is the probability of correctly answering item j, given that the examinee is a member of class t. This expression thus allows the expression of the manifest probability in terms of the latent probabilities. Conceptually then, LCA separates data into latent classes using the criteria that in the resulting classes, the manifest joint probabilities, that is f_{ijkl}, should equal the product of the class probability and the class conditional probabilities for each item, as in (3).

Estimation of Latent Parameters – MLE

As mentioned above, determining the proportions correct of each item within each class amounts to solving a system of equations or maximum likelihood estimation (MLE). Though a detailed discussion of MLE is outside the scope of this thesis, a brief description of how the method works is instructive, particularly as it applies to LCA. In general, MLE is used to derive the value of unknown parameters from a set of observed values. To take a simple example, one could use maximum likelihood to determine the most likely values for the mean and variance (latent parameters) of a population given certain sample values (observed scores). Two methods are generally used to derive these values; the first is a closed solution based on differential calculus and the second is an iterative numerical estimation procedure called expectation maximization ([EM], Dempster, Laird, & Rubin, 1977). The first approach requires that the first derivative be taken of the likelihood function, that is, the function describing the likelihood of particular latent parameters estimates being the true values given the observed data. The first derivative provides the value of the slope of the function at the parameter values. Since the maximum of the function will always have a slope of zero, the first derivative is

set to zero and the values of parameters in the resulting equation are determined in the solution of the equation. In the case of complex functions with many variables the iterative EM approach is used. In the approach pioneered by Goodman (1974), starting values for the parameters of the latent classes (i.e., class probabilities [$\pi_X(t)$] and class conditional probabilities for each item [$\pi_j|x(t)$]) are provided in order to calculate the expected values of the manifest variables. In turn, the values of the latent parameters are re-calculated from both the expected and observed values of the variables, the so-called maximization step. This process continues for a number of iterations until each subsequent iterative step produces no significant change in the solution. Goodman (1974) shows that, provided a solution exists, this method will converge on the maximum likelihood estimates for the latent class model. Of course, the fit of the MLEs must then be assessed using methods mentioned previously. [3]

To summarize, the basic steps taken in a LCA analysis are as follows. First, data from a number of indicator variables is collected. Second, a series of candidate LC models are specified. Next, MLE estimates are generated for each model and analyzed for their fit to the observed data using χ^2 criteria, and then using L^2 to compare fit among competing models. Once a model has been chosen, respondents can be classified to states identified in the model by choosing the state to which they most likely belong, given their response to the test items.

---

[3] Technically, K-means cluster analysis is also a form of expectation-maximization. The expectation step classifies observations to clusters based on their closeness to the interim cluster centres. The maximization step is the recalculation of the cluster centre from the mean of all observations assigned to the cluster.

Table 3.5. Observed data presented in Goodman (1974).

Row	Item				Observed Frequency
	A	B	C	D	
1	+	+	+	+	42
2	+	+	+	-	23
3	+	+	-	+	6
4	+	+	-	-	25
5	+	-	+	+	6
6	+	-	+	-	24
7	+	-	-	+	7
8	+	-	-	-	38
9	-	+	+	+	1
10	-	+	+	-	4
11	-	+	-	+	1
12	-	+	-	-	6
13	-	-	+	+	2
14	-	-	+	-	9
15	-	-	-	+	2
16	-	-	-	-	20
Totals	171	108	111	67	216
p-values	.792	.500	.514	.310	1.0

LCA demonstrated: Goodman (1974)

In order to see how the above steps work in a real example, let us take a moment to review the analysis procedure by Goodman (1974). In Goodman's example, the data are from 216 survey respondents on 4 dichotomous items all assumed to measure the same dimension. The data for all respondents are presented in Table 3.5. For the sake of simplicity, the present demonstration will focus on 3 competing LC models: (a) a one-class baseline model, (b) a two-class unrestricted model and, (c) a two-class restricted model imposing the constraint that certain class conditional probabilities be equal.

Since the models have been specified, the next step in LCA is to derive MLE estimates for all the latent parameters in each model. For the one class model, no parameters will need to be estimated since these parameters will not vary from class to class. Rather, the "class conditional" parameters are simply the p-values for each item. Thus, the chi-squared test is simply a test of the difference between the observed joint probabilities (i.e., the observed frequencies from Table 3.5 divided by 216) and the product of the item probabilities corresponding to the response pattern in each row of Table 3.5. For example, the estimated probability of the fifth row based on the one class model equals 0.792 x (1 - 0.500) x 0.514 x 0.310, or 0.063. Notice that the observed probability for the fifth row is 6 / 216, or 0.028 indicating that for this row at least, the one class model appears not to fit well. The χ^2 (10, $\underline{N}$ = 216) value for this model is 104.7, $\underline{p}$ <0.001 and thus, the one class model overall provides a poor fit to the observed data.

Table 3.6. Class Probabilities and Class Conditional Probabilities for Each Item for both Restricted and Unrestricted Two Latent Class Models

	Latent Class	$p(X_t)$	$p(A_1\|X_t)$	$p(B_1\|X_t)$	$p(C_1\|X_t)$	$p(D_1\|X_t)$
Unrestricted	1	.279	.993	.940	.927	.769
	2	.721	.714	.330	.354	.132
Restricted	1	.279	.993	.933	.933	.771
	2	.721	.732	.342	.342	.132

For the two-class unrestricted model, the parameters first must be calculated using MLE. The parameters so estimated are presented in Table 3.6. Examining the class conditional probabilities for each item, it appears that the first class represents those respondents that responded 'correctly' to each of the test items whereas respondents assigned to Class 2 represented those that responded 'incorrectly'. Examining the fit of this model can help determine whether these two latent classes account for the response data. To calculate the χ^2 value for this model, one compares the same observed joint probabilities from the first model with the products of the class probability and the class conditional probabilities for each item summed for each class. Note that this precisely reflects the definition of local independence. Within a given class, the probability of a given response pattern is the product of the probabilities for each individual item. The predicted probability for the fifth row given the two class unrestricted model is:

(0.279 x 0.993 x [1 - 0.940] x 0.927 x 0.769) + (0.721 x 0.714 x [1 – 0.330] x 0.354 x

0.132), or 0.028, an exact match to three digits of the observed probability. The χ^2 value reflecting the overall fit of the two class model is χ^2 (6, $\underline{N}$ = 216) = 2.720, $\underline{p} > 0.05$ indicating very good model-data fit.

The last model to be assessed is a restricted two-class model, reflecting the expectation that the class conditional probabilities for some indicators will be equal. The specific restrictions imposed for this model was the equivalence of $p(B_1|X_1) = p(C_1|X_1)$ and also $p(B_1|X_2) = p(C_1|X_2)$. This restriction has the effect of limiting the number of parameters to be estimated increasing the degrees of freedom and consequently is a more parsimonious model than the two-class unrestricted model. The MLE values for this model are presented in the bottom half of Table 3.6. As can be seen, the restrictions make little absolute differences to the values in the unrestricted two-class model and thus, little difference to the chi-square statistic is expected. This was indeed the case, χ^2(8, $\underline{N}$ = 216) = 2.838, $\underline{p} > .05$. With regard to the comparison in fit between the two-class restricted and unrestricted models, the difference between the likelihood ratio statistics for each model is examined. Doing this, it is found that L^2(2, $\underline{N}$ = 216) = 0.166, $\underline{p} > .05$, indicating no significant difference in model-data fit. Thus, considering the greater parsimony of the restricted model and no reduction in fit, the restricted two-class would be chosen.

An application of LCA to Educational Data

Haertel (1989) defined a restricted class model known as the *binary skills model* in an attempt to determine (a) whether data from a reading achievement test could be adequately described using latent classes and if so, (b) what was the skill structure underlying the test. Haertel applied the binary skills LC model to the analysis of

responses to 37 items from a reading comprehension test. Because LCA requires that a tabulation of each possible response pattern be provided, the number of items that can be analyzed is limited. Therefore, Haertel analyzed 18 subsets of 6 items each and determined the parameters of best fitting model for each subset. When items were consistently identified as belonging to the same latent class they were gathered together into clusters. This process continued until a skill map could be identified across the entire test. The final step in the process was the substantive interpretation of each latent class.

Haertel (1989) showed that 28 of the 37 items were clearly associated with one of five latent classes. That is, items for which performance was similar could be grouped together into subsets, thus leading to the formation of a class structure of the test. Subsequently the items that defined each cluster were analyzed to determine if there were skills that were common to all. Skills were identified by Haertel, two of which corresponded to particular reading passages in the test, with the remaining three described as sentence comprehension, inference, and sophisticated inference. Furthermore, the clusters themselves and the items that composed them were ordered such that mastery of a particular class implied the mastery of all previous classes. This suggested that, at least for 28 of the items, states of competence were ordered unidimensionally, and therefore the test had a unidimensional structure.

The preceding research demonstrates the key advantages of latent class analysis for the determination of test structure. First, Haertel (1989) showed that performance on a significant portion of the test can be well described by a small number of latent classes. Second, the classes appear to be ordered, and therefore an overall structure for the test

can be specified. In this case, the overall structure appeared to be unidimensional. Last, the classes themselves can be examined to determine what types of item responses comprised them, and therefore what a description might be in terms of test content.

Several challenges with using LCA for test structure analysis are worth noting. The first challenge involves the choice of LCA model. Like K-means cluster analysis, the number of classes must be specified for an analysis to be conducted, but the addition of constraints makes the number of candidate models for consideration large, particularly when the number of items and potential latent classes is large. For example, Lindsay, Clogg, & Grego (1991) demonstrated that under a strict set of assumptions, latent class models needed approximately half the number of classes as test items to adequately model simple item characteristic curves. In a more exploratory framework, determining the adequate number of classes with appropriate set of constraints could prove intractable. In fact, Uebersax (2001) demonstrates that local maximum solutions are more likely as the number of latent classes in the model increases. Consequently, the use of LCA when the number of potential classes is large is questionable.

In summary, Latent Class Analysis is a method by which parameters of hypothetical latent class models can be derived from observed data. Since the fit of these parameters for the observed data can be assessed statistically, LCA allows the determination of the best fitting of a number of candidate models. For educational test data, a well-fitting model could be considered equivalent to identifying the test's structure.

Summary

From the above review of latent state methods, it is clear that such accounts can provide key information about test structure not available from latent trait methods. In

particular, these methods may reveal specific characteristics of both the individual test taker and of the states of competence underlying test performance as a whole. This type of information may be particularly useful when categorical judgments are required from test performance such as standard setting. However, the type of information provided by latent state test structure methods is not in conflict with those from dimensional account. Rather, the types of information from the two classes of methods are complementary. Indeed, dimensional accounts may provide a kind of organizing framework for latent states. In the next chapter, a class of analytical tools known as artificial neural networks is described. It will be argued that these networks may have application in educational measurement as a method for determining test structure incorporating aspects of both dimensional and latent state accounts.

References

Beller, M. (1990). Tree versus geometric representation of tests and items. *Applied Psychological Measurement, 14*, 13–28.

Clogg, C. C. (1995). Latent Class Models. In G. Arminger, C. C. Clogg, & M. E. Sobel (Eds.), *Handbook of statistical modeling for the social and behavioral sciences.* New York: Plenum.

Corter, J. E. (1995). Using clustering methods to explore the structure of diagnostic tests. In P. Nichols, S. Chipman, & R. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 305–326). Hillsdale, N J: Erlbaum.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*, 1-38.

Everitt B. S. & Hand D. J. (1981). *Finite mixture distributions*, London: Chapman & Hall.

Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika, 61*, 215-231.

Goodman, L.A. (1979). Simple models for the analysis of association in cross-classifications having ordered categories. *Journal of the American Statistical Association, 74*, 537-552.

Haertel, E. (1989). Using restricted latent class models to map the skill structure of achievement items. *Journal of Educational Measurement, 26*, 301–321.

Lazarsfeld, P. F. & Henry, N. W. (1968). *Latent structure analysis.* Boston: Houghton Mifflin.

Lindsay, B., Clogg, C. C., & Grego, J. (1991). Semiparametric estimation in the Rasch model and related exponential response models, including a simple latent class model for item analysis. *Journal of the American Statistical Association, 86*, 96-107.

McCutcheon, A. L. (1987). *Latent class analysis.* Beverly Hills, CA: Sage Publications.

Milligan, G.W. (1981). A monte carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika, 46*, 187-199.

Mislevy, R. J., Almond, R. G., Yan, D., & Steinberg, L. S. (2000). *Bayes nets in educational assessment: Where do the numbers come from?* (Technical Report no. 518). Los Angeles, California: California University, Center for the study of evaluation.

Roussos, L. A., Stout, W. F., & Marden, J. I. (1998). Using new proximity measures with hierarchical cluster analysis to detect multidimensionality. *Journal of Educational Measurement, 35*, 1-30.

Sireci, S. G., (1995, August). *Using cluster analysis to solve the problem of standard setting.* Paper presented at the annual conference of the American Psychological Association, New York, NY.

Sireci, S. G., Robin, F., & Patelis, T. (1999). Using cluster analysis to facilitate standard setting. *Applied Measurement in Education, 12*, 301-325.

Stout, W., Habing, B., Douglas, J., Kim, H.R., Roussos, L., & Zhang, J. (1996). Conditional covariance based nonparametric multidimensionality assessment. *Applied Psychological* Measurement, *20*, 331-354.

Tay-lim, B. S. & Stone, C. A. (2000, April). *Assessing the Dimensionality of Constructed-Response Tests Using Hierarchical Cluster Analysis: A Monte Carlo Study*. Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans, LA.

Uebersax, J. (2001). *A Brief Study of Local Maximum Solutions in Latent Class Analysis.* http://ourworld.compuserve.com/homepages/jsuebersax/local.htm

Chapter 4 – Artificial Neural Networks

Thus far, two general classes of methods to determine test structure have been reviewed. The first methods, those reviewed in Chapter 2, described test structure in terms of a small number of continuous dimensions. The descriptions generated from these methods were based upon statistical relationships between performances on test items, and presumed that variations in examinee performance were a result of variations in the possession of a latent psychological trait.

The methods in Chapter 3 presumed that the essential characteristics of test performance are captured in examinee membership in one of a finite number of categories, or classes. The classes identified by these methods could then be examined to determine the specific characteristics that defined them, and the relationships among the classes could shed light on the overall structure of the test.

The goal of this chapter is to motivate the investigation of artificial neural networks (ANNs) as a third method to identify test structure. It will be argued that a type of network known as the Self-Organizing Map ([SOM], Kohonen, 1982, 2001) is particularly appropriate for this purpose because it has the potential to combine many of the advantages of both latent trait and latent state accounts of test structure. To demonstrate this, the chapter is organized as follows. First, an overview of ANNs will be provided in order to acquaint the reader with the general classes of networks and how they work. Then, the SOM will be described in detail, focusing on the assumptions that underlie its use and its capacity both to create latent trait and latent state descriptions of data. Last, an application of SOMs in educational measurement will be reviewed. As a result of this review it will be argued that an empirical study, focused on variables of

known importance to educational measurement, is needed to determine the conditions under which SOMs are appropriate for determining test structure.

What are ANNs?

ANNs are functional models of some essential features of neural processing (Rumelhart, McClelland, & the PDP Research Group, 1986). They consist of many simple processing elements, analogous to neurons, whose function is to sum incoming 'activation,' mathematically transform it, and then propagate the result to other processing elements. The level of activation contributed to an element by a single connection is determined by the activity of the element preceding the connection multiplied by a *weight*. It is the value of the weights in the network that determines the relationship between inputs and outputs.

The advantages of these networks for educational measurement are twofold: computational power and the ability to abstract interpretable relationships between variables. More specifically, ANNs learn to create connections between large numbers of input and output variables based on mathematical and statistical relationships between them. The nature of this relationship can then be determined by examining the structure of the network (e.g., Carbonaro, 2003; Dawson & Zimmerman, 2003; Leighton & Dawson, 2001). Note that this is analogous to what all methods that determine test structure do: extract the fundamental relationships between item-level performance (input) and test-level inferences (output) which are then examined for their structure. This similarity suggests that ANNs may be a useful method to determine test structure.

To better conceptualize what ANNs are, it is helpful to consider them in relation to a standard statistical procedure, linear regression. Figure 4.1 shows how regression

would be represented as an ANN. As can be seen, the network comprises two separate 'types' of units: those representing independent variables (input units) and those representing dependent variables (output units). In network terminology, the input units (independent variables) are multiplied by their respective connection weights (regression coefficients), summed by the net input function, and transformed by the linear activation function to determine the activation of the output unit (the dependent variable).

Figure 4.1. Neural Network Representation of a Linear Regression Equation

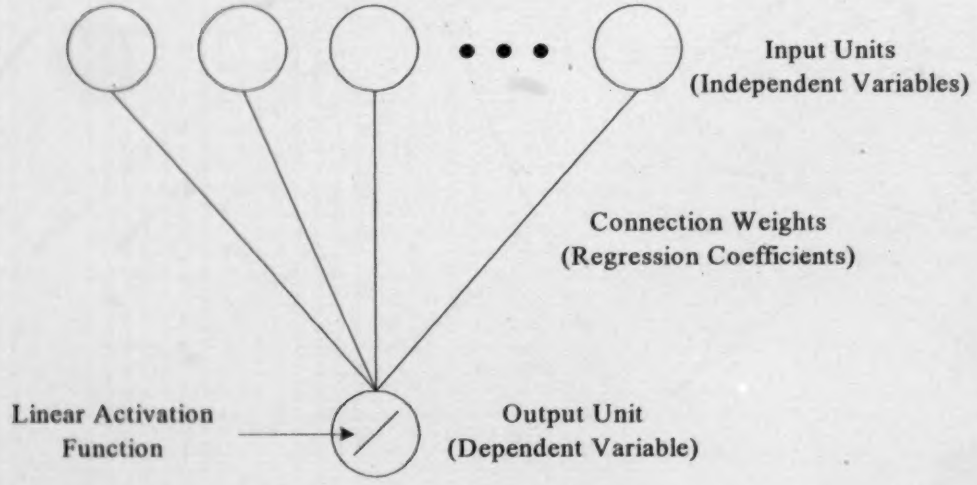
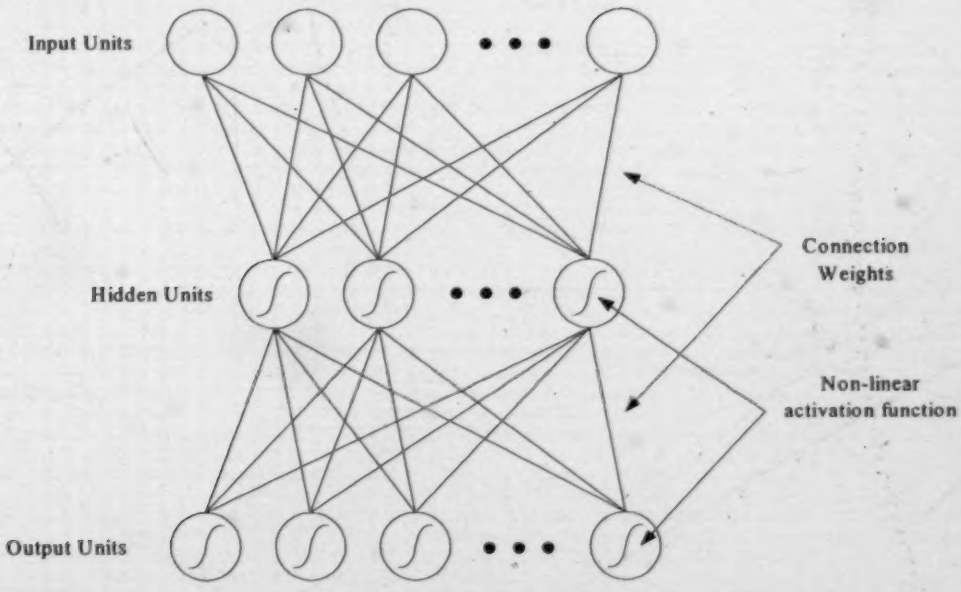


Figure 4.2. Three Layer Artificial Neural Network



Networks most often differ from linear regression in several essential ways (see Figure 4.2). First, neural networks often feature an additional layer of units between input and output known as *hidden* units. These units are considered feature detectors and are necessary when the relationship between inputs and outputs is complex, for example when simultaneous activation of two or more units is necessary to 'activate' an output unit. Second, the activation of a given unit is not restricted to a linear function of its inputs. The function can also be non-linear, for example, logistic (e.g., Rumelhart, Williams, & Hinton, 1986) or Gaussian (e.g., Dawson & Schopflocher, 1992). Last, in a regression equation, the value of the dependent variable is known when the regression weights are calculated. This is not a precondition for all ANNs. As shall be seen in the next section, *supervised* networks require the value of the output whereas *unsupervised* networks do not.

Supervised ANNs

As mentioned above, for supervised networks the target output is known a priori for each input pattern. In the process of learning, this type of network calculates the difference between the target and actual output then adjusts the values of its weights over many iterative steps to minimize this difference. This compares to linear regression analysis where the difference between the predicted and observed values of the dependent variable is minimized by finding suitable values for the regression weights. The result for a supervised network is that, given sufficient computational power (typically, a sufficient number of hidden units), the ANN can produce the correct output for a given input.

Generally, there are two types of applications for supervised networks in educational measurement, generalization of rules and problem analysis. In applications involving generalization, a network is trained using a subset of possible input data. The ability to predict the value of the outputs (dependent variables) from input data not in the training set tests the generality of the relationship abstracted by the network. When prediction is good, the network is considered to have learned relationships that relate to the domain as a whole rather than to the specific instances in the training set. Examples of this kind of research in educational measurement include the prediction of item difficulty from item features (Carbonaro, 2003; Perkins, Gupta, & Hammana, 1995).

In applications involving problem analysis, connection weights and hidden unit activations of a trained network are analyzed to determine what features or combinations of features in the input layer predict certain outcomes. This is analogous to determining which independent variables are statistically significant in a regression equation. Applications relevant to educational measurement include the derivation of knowledge

states in Tatsuoka's (1983, 1995) rule-space model from the Q-matrix (Hayashi, 2003), determining the computational resources required to generate responses in the Wason (1966) reasoning task (Leighton & Dawson, 2001), and discovering rules that could govern performance on the balance scale task (Dawson & Zimmerman, 2003).

In summary, the advantages of supervised networks lie in their ability to learn associations between inputs and outputs when the target outputs are known. Specific advantages are the capability of classifying input patterns into categories by abstracting and generalizing rules. Also, supervised ANNs can help identify the specific features that predict the outcome, thus providing greater insight into the nature of the problem being solved by the ANN. However, in the exploratory determination of test structure, the output depends upon whatever structure is inherent to the data and therefore the target for that output is not known. In this situation, unsupervised networks, discussed next, may prove to be helpful.

Unsupervised ANNs

In contrast to the supervised case, unsupervised networks serve the function of *uncovering* the structure of the data, reflecting the probability density of the inputs (e.g., Kohonen, 2001; Hinton & Sejnowski, 1999; Rumelhart & Zipser, 1985). Since the target output is not known for each set of inputs, the goal of unsupervised networks is to *assign* an output to a given input pattern. In general, it is the similarity among input patterns that determines the network output. This property makes the processing from unsupervised ANNs conceptually similar to cluster analysis.

Unsupervised ANNs work by allowing output units to *compete* for activation. That is, only the connection weights associated with the outputs unit(s) of highest activation

when a given input pattern is present will change. Since different input patterns will activate different output units, the effect of training is to make specific output units selectively sensitive or *tuned* to particular input patterns. When there are an equal number of unique input patterns and output units, each output unit could become tuned to a single pattern. When the number of input patterns exceeds the number of output units, units will become tuned to *clusters* of patterns such that similar patterns will tend to activate the same output unit. It is in this way that unsupervised networks come to reflect the structure of the input patterns.

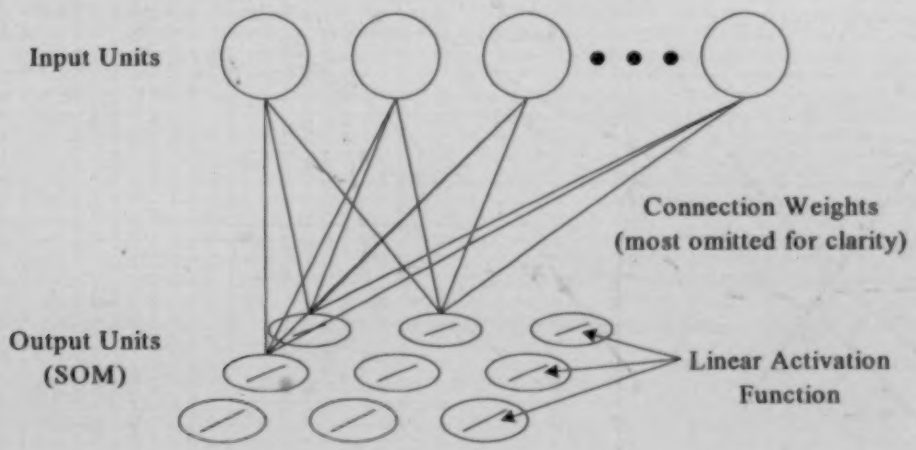
In the next section, a type of unsupervised ANN will be reviewed that could provide specific information regarding not only the cluster structure of the data, but also what the arrangement of the clusters reveal about test structure. This ANN is known as the self-organizing map (SOM).

Kohonen's SOM

The SOM is an unsupervised ANN with several unique features, namely, that it produces a map-like representation of the input patterns that is both ordered and weighted by frequency (Kohonen, 1982, 1990, 2001). A representation of a SOM is presented in Figure 4.3. The SOM is map-like in the sense that the structure of the data is projected on a lattice of output units, usually in two dimensions. Each point of the lattice has a corresponding *model vector*, essentially the geometric mean or *centroid* of all input patterns that activate it. The model vectors essentially play the role that weights play in other neural network models; they provide a means by which each element in the input vector biases the selection of best-fitting unit at the subsequent layer, in this case, units of the SOM. The lattice is ordered in the sense that the model vectors, "tend to attain values

that are ordered along the axes of the network" (Kohonen, 1990, p. 1467). Though this ordering is difficult to formalize in a network for which there are many input units (i.e., when input vectors are multidimensional), for the case in which input patterns are scalars and the output layer (i.e., the SOM) is a one-dimensional array of points, the values corresponding to the points are ordered sequentially, either descending or ascending (Kohonen, 1982). Last, the SOM is weighted by frequency in that similar observations that occur more frequently in the dataset are allocated more space in the map. In this way, the SOM could be considered to reflect the *probability density* of the input patterns.

Figure 4.3. Architecture of the Kohonen Self-Organizing Map



How might the essential properties of the SOM, that it produces a map-like, ordering of model vectors (centroids) that reflect probability density, be useful for the determination of test structure? First, like latent trait accounts of test structure, the map-like ordering of model vectors could reveal a dimensional facet to test performance. By determining the basis for the ordering of vectors in the SOM, both the number of

dimensions and the substantive interpretation of them may be revealed. Second, like latent state accounts of test structure, the model vectors themselves could be examined to determine their characteristics in terms of item- or subtest-level variables. The characteristics may help define the most prevalent latent states of examinees.

Table 4.1. Interpreting Model Vectors from a 2 x 2 SOM

	X-Coordinate	
Y-Coordinate	1	2
1	(1, 1, 1, 1)	(1, 1, 0, 0)
2	(0, 0, 1, 1)	(0, 0, 0, 0)

A simple example of a possible SOM representation of test structure is revealed in Table 4.1. In this table, each cell represents a location in the SOM, and a possible state of mastery on a hypothetical test, say a unit test on Chinese geography and culture. By examining the organization of the map, it can be noted that the diagonal from (1,1) to (2,2) could be seen as representing some continuous dimension of overall ability on the test. The diagonal orthogonal to the first [from (1, 2) to (2, 1)] could represent deviations from expected performance given overall ability and therefore may suggest that the test is manifestly two-dimensional. By analyzing the model vectors in each cell, insight into the nature of examinee performance giving rise to these model vectors can be gained. Let's assume that the first two items require knowledge of Chinese geography and the last two require knowledge of Chinese culture. A possible interpretation of the second dimension would therefore be that it represents selective ability in either culture or geography.

The mechanics of self-organization

Training a SOM consists of positioning model vectors so as to minimize the difference between each input vector and its associated model vector. This difference is usually operationalized in terms of the Euclidian distance between vectors[4], i.e.,

$$= \sqrt{(x_1 - m_{c1})^2 + (x_2 - m_{c2})^2 + \ldots + (x_n - m_{cn})^2}, \qquad (4.1)$$

where x_n is the n^{th} element of the input vector $\mathbf{x} \in \mathbf{R}^n$ and m_{cn} is the n^{th} element of the model vector $\mathbf{m}_c$ that most closely matches $\mathbf{x}$, that is, the winner. When a set of model vectors is found that minimizes the above distance for all input vectors simultaneously, these model vectors form a representation of the cluster structure in input space in a resolution defined by the number of units in the output layer (i.e., the size of the SOM).

How do the model vectors attain these best-fitting values? First, because the algorithm underlying the SOM is iterative, small adjustments to the value of the model vectors are made following many presentations of the input patterns. However, for a given presentation of an input, only the model vectors that most closely match the input vector are updated. Note that the Kohonen SOM updates not only the winning output unit but also those in proximity to it, that is, in the winning unit's neighbourhood. This characteristic will be discussed in detail in the following section. The winning vectors are updated according to the following rule:

---

[4] The inner or dot product has also been used to define the differences, or more precisely, the similarity between input and model vectors. This method more closely resembles the net input function in most other neural network models. In this case, normalization of all vectors is necessary after each updating step, greatly increasing the computational requirements of the SOM. For the purposes of the present paper, only the SOM that defines differences in terms of vector norms will be reviewed.

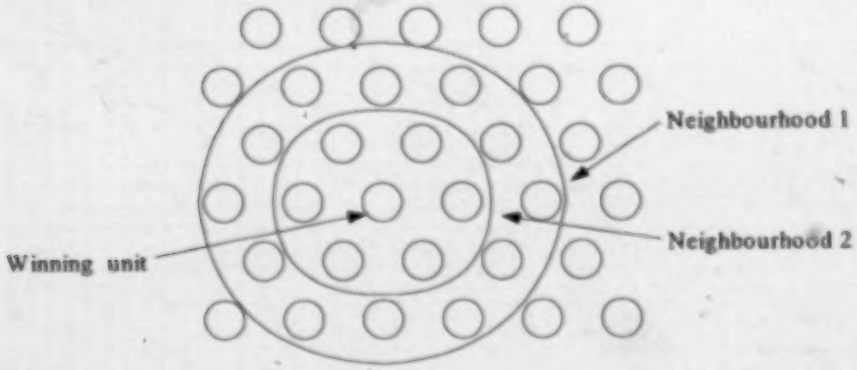
$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)] \quad (4.2)$$

where t is a measure of time, or the step in the iterative process, and $\alpha(t)$ is a scalar, $0 < \alpha(t) < 1$ that decreases monotonically with increasing t. Equation 4.2 has the effect of moving the winning model vectors 'closer' towards the input vector. All other model vectors remain unchanged. Because $\alpha(t)$ decreases over time, the magnitude of change in the winning model vectors tends to decrease as the network learns. This is done because as the SOM becomes more finely tuned to the data, only small refinements are necessary to improve its precision. Large changes later in learning are undesirable because they may lead to 'over-shooting' of the ideal correction. Finally, two choices exist for the timing of updating the model vectors, (a) following the presentation of each individual input pattern, or (b) following the presentation of a set of patterns. The latter case avoids the unnecessary fluctuation of model vectors due to unsystematic variation in the input vectors.

As mentioned above, a feature unique to Kohonen networks is that model vectors in a neighbourhood of m_c, the winning unit, are all subject to change in the direction of the input vector x while vectors outside this neighbourhood are left unchanged. The neighbourhood N_c is defined in terms of the size of the radius around m_c. Figure 4.4 shows two such neighbourhoods defined in terms of unit of the SOM; all units within the circular boundaries constitute the neighbourhood. The purpose of the neighbourhood is to facilitate the ordering of the model vectors over the surface of the SOM. Kohonen (1990) states that by making the initial radius of the neighbourhood large (up to one-half of the radius of the entire SOM), the network learns a coarse coding of the input space.

Then, by subsequently decreasing the size of N_c as the network learns, individual regions of the SOM become more finely tuned to corresponding regions of input space but the entire SOM retains its global ordering.

Figure 4.4. Two Neighbourhoods Defined on a SOM.



The amount that model vectors in the neighbourhood are changed as a result of learning depends on the choice of neighbourhood function. Two functions suggested by Kohonen (2001) are the (a) binary and (b) Gaussian functions. The binary function simply applies the same $\alpha(t)$ from equation (2) for all model vectors in the neighbourhood of m_c. No changes are made to model vectors outside the neighbourhood. In the Gaussian case, a function $h_{ci}(t)$ is substituted for $\alpha(t)$, for example,

$$h_{ci}(t) = \alpha(t) \cdot \exp\left(\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right) \qquad (4.3)$$

where r_c is the location of m_c and r_i is the location of some other model vector, m_i. $\|r_c - r_i\|^2$ is thus the distance between m_c and m_i and σ is the parameter that determines the range over which the neighbourhood function is operating. Equation 4.3 thus defines a neighbourhood function whose effect diminishes smoothly as distance from m_c increases.

Using the SOM

The following are the steps that could be followed in using the SOM to analyze data. First, the parameters of the network must be established. This includes choosing (a) the size of the map and its respective dimensions, (b) the measure of difference between input and model vectors, (c) the neighbourhood function, (d) a starting value for α, (e) the rate of decay for both the size of the neighbourhood and the value of α, (f) the number of cycles between updating the model vectors, and (g) the number of cycles over which the network will run. (See Kohonen, 2001, for rules of thumb regarding the settings of these parameters.) Next, the model vectors are set to random values in [0, 1] and the inputs are presented successively to the network. After the network runs for the designated number of cycles, the network can be analyzed to determine the values of each of the model vectors. This analysis will help determine both the cluster structure of the data and its organization.

Applications of SOMs in Educational Measurement

SOMs have been employed in several areas in educational measurement. These include automated scoring (Williamson & Bejar, 2000; Williamson, Bejar, & Sax, 2004), on-line assessment in computer-based tutorials (Mullier, 2003), and in assessment of complex performance tasks (Kanowith-Klein, Stave, Stevens, & Casillas, 2001; Stevens, Ikeda, Casillas, Palacio-Cayetano, & Clyman, 1999; Stevens, Johnson, & Soller, 2005;

Vendlinski & Stevens, 2002). These studies have focused primarily on the capacity of SOMs to differentiate between qualitatively different types of solution strategies on complex problems.

To take an example, Stevens et al. (1999) used SOMs to analyze the development of competence in computer-based performance assessments. The performance assessments examined in this study were unstructured problems in two disparate domains, high school genetics and case-based medical reasoning. Each of these problems requires the examinees to request information that they believe would lead them closer to solution of the problem. For example, in the high school genetics problem called *True Roots*, the examinees' task is to request specific genetic tests to help determine whether two babies had been mistakenly switched at birth. Optimal use of the information provided by the tests required both an understanding of the genetic principles underlying the tests as well as an ability to use the information to move closer to the goal state.

Of primary interest to Stevens et al. were the paths to solution exhibited by the examinees. That is, what information did examinees request in the process of solving the problem and when did they request it? Information requests were represented in a vector having one element for each unique piece of information. Stored in this vector was whether or not each piece of information was requested and the order in which requested information was called for. In particular, each element stored the next-requested piece of information. For example, if an examinee requested consecutively the 4^{th}, 10^{th}, and 13^{th} pieces of information, the 4^{th} element of the examinee response vector would contain the number 10, and the 10^{th} element would contain the number 13. In this way, the entire

solution path could be represented in the vector and the choice of solution path (i.e., identity and order of information requests) could be faithfully represented.

The Kohonen net developed for each problem had one input unit for each element in the vector. The size of the output layer, that is, the map itself varied between the two applications, 5 units by 5 units for the *True Roots* problem and 10 by 10 for the medical case study. The sizes were informed by the complexity of the respective problems. After examinee solutions were obtained for each of the problems, the network was trained.

For the *True Roots* problem, the key research questions were to determine how competence developed over time and whether the SOM could represent qualitative differences between states of mastery. The training set for this problem consisted of solutions from undergraduate students (n = 156). Since most examinees solved the *True Roots* problem it was anticipated that important differences in performance would be captured by the type and identity of information requests in their solution. In order to identify the most effective solutions, each examinee performed the task up to three times. The assumption was that later attempts would be of higher quality than earlier ones. The analysis of differences between attempts showed that several types of sub-optimal strategies characterized early attempts, such as the reliance on only one type of information (e.g., pedigree but not blood type data) or exhaustive but unsystematic information gathering. Later attempts showed a much more targeted approach with few redundant or superfluous information requests.

Grade 10 students, for whom the *True Roots* problem was designed, were also categorized using the SOM trained with data from undergraduates. Not surprisingly, the majority of their first solutions activated output units indicating inefficient performance.

Subsequently, these students were given specific instruction in critical thinking and problem solving strategies. Though no information regarding the performance of a control group was provided, the percentage of these students correctly solving the problem increased, from 33% to 64%. Most interestingly, the output units activated by the data from later performances were the same units that indexed more advanced problem solving ability in undergraduates.

The medical case assessment was designed as part of a battery that tested the readiness of would-be doctors to practice medicine unsupervised. As such, the case study was much more complex than the *True Roots* problem, having over 2300 possible information requests. One hundred randomly sampled performances served as the training set for the SOM. After training, 20 clusters were identified and each was scored according to pre-established criteria for the problem. The National Board of Medical Examiners (NBME) set these criteria using subject matter experts (see Clauser, Subniyah, Nungester, Ripkey, Clyman, & McKinley, 1995, for detail on the establishment of the performance criteria). Interestingly, it was found that model vectors associated with high quality solutions had the greatest number of examinees classified to its corresponding cluster. This concordance of quality with frequency suggests that a small number of specific solution paths are associated with developed competence and conversely, and that a larger number of others are characteristic of pre-competence states. In this problem, and likely in many others, there are few ways to perform well and many ways to perform poorly.

The research by Stevens et al. (1999) highlights some of the characteristics of SOMs in determining test structure. Their work shows that examinee solution paths that

are similar tend to be assigned to a similar location in the SOM. These locations appear to represent latent states of competence in the problem-solving domain. Furthermore, the model vectors associated with these locations were analyzed to determine the characteristics of the responses that lead to their activation. In the Stevens et al. case, this type of analysis led to the identification and order of information requests in the *True Roots* and medical reasoning problems. When this analysis was conducted for each location in the map, it defined a kind of population of competence states for the assessment.

Second, the SOM representation of examinee performance was interpreted in terms of the quality of their responses. In particular, the SOM solution was combined with other analyses, for example, comparison of high school examinee performance with undergraduates, using pre-established criteria to describe SOM locations in terms of the quality of medical reasoning. This allowed interpretation of examinee performance in terms of the content domain. One potential advantage germane to the application of SOMs to determining test structure is the capacity of these networks to impose order on the set of model vectors. Ideally, this order would represent the *quality* of solutions in the SOM, and could be compared to and corroborated with other criterion measures. However, it appears as though the specific characteristics of the examinee data in this problem-solving environment did not permit such ordering. Specifically, geometrically similar solutions were not always similar in quality.

Third, examinees' performances were associated with specific ability states identified by the SOM. That is, the characteristics of the cluster in terms of solution quality were ascribed to the examinee who was assigned to that cluster. This is

tantamount to the SOM informing a scoring model for performance on the task. Quality of task performance was assessed based on the comparison of the locations activated from a given performance and performances of other examinees whose ability was known.

Stevens' work is revealing about the utility of SOMs to analyze the development of competence on complex tasks, but also highlights specific questions with respect to the identification of test structure. That is, the use of the SOM in their work was innovative, but left questions about general principles regarding its appropriate use. First, one of the central characteristics of the SOM is its capacity to create an ordered representation of data that mirrors the probability density of the data set. However, the SOM in Steven's work did not appear to be ordered in any meaningful way. An important question when using SOMs for educational measurement is therefore, what characteristics of the data are necessary to create an ordered, interpretable representation? A second related issue concerns the characteristics of the SOM. It was mentioned that two sizes of SOMs were used, 5 x 5 for the *True Roots* problem and 10 x 10 for the more complex medical reasoning problem. No precise rationale for this decision was provided. Without a systematic investigation of the conditions that support the creation of SOMs with desired characteristics and the robustness of the SOM with respect to deviations from these conditions, inferences derived from their use may be inappropriate or misleading. In particular, the precise characteristics of the data and the configuration of the SOM are of central importance. Therefore, an important question for applications of SOMs in educational measurement is to determine what those characteristics and configurations are, and what the potential consequences are of not choosing maps with appropriate

characteristics. Clearly, these issues must be systematic addressed if a balanced evaluation of the utility of SOMs for educational measurement is to be made.

The remainder of this thesis describes research executed to accomplish such an evaluation.

References

Carbonaro, M. (2003). Making a connection between computational modelling and educational research. *Journal of Educational Computing Research, 28*, 63-79.

Clauser, B. E., Subhiya, R. G., Nungester, R. J., Ripkey, D. R., Clyman, S. G., & McKinley, D. (1995). Scoring a performance-based assessment by modeling the judgments of experts. *Journal of Educational Measurement, 32*, 397-415.

Dawson, M. W. & Schopflocher, D. P. (1992). Modifying the generalized delta rule to train networks of nonmonotonic processors for pattern classification. *Connection Science, 4*, 19-31.

Dawson, M.R.W. & Zimmerman, C. (2003). Interpreting the internal structure of a connectionist model of the balance scale task. *Brain and Mind, 4*, 129-149.

Hayashi, A. (2003, April). *A comparison study of rule space method and neural network model for classifying individuals and an application.* Paper presented at the annual meeting of the National Council on Measurement in Education. Chicago, IL.

Hinton, G. & Sejnowski, T. J. (1999). *Unsupervised learning: Foundations of neural computation*. Cambridge, MA: MIT Press.

Kanowith-Klein, S., Stave, M., Stevens, R., & Casillas, A. M. (2001). Problem-solving skills among precollege students in clinical immunology and microbiology: Classifying strategies with a rubric and artificial neural network technology. *Microbiology Education, 2*, 25-33.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*, 59-69.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE, 78*, 1464-1480.

Kohonen, T. (2001). *Self-organizing maps* (3rd ed.). Berlin: Springer-Verlag.

Leighton, J. P. & Dawson, M. R. W. (2001). A parallel distributed processing model of Wason's selection task. *Cognitive Systems Research, 2*, 207-231.

Mullier, D. (2003). A tutorial supervisor for automatic assessment in educational systems. *International Journal on E-Learning, 2*, 37-49.

Perkins, K., Gupta, L., & Tammana, R. (1995). Predicting item difficulty in a reading comprehension test with an artificial neural network. *Language Testing, 12*, 34-53.

Rumelhart, D. & Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science, 9*, 75-112.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1: Foundations). Cambridge, MA: MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propogation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1: Foundations). Cambridge, MA: MIT Press.

Stevens, R., Ikeda, J., Casillas, A., Palacio-Cayetano, J., & Clyman, S. (1999). Artificial neural network-based performance assessments. *Computers in Human Behavior, 15*, 295-313.

Stevens, R., Johnson, D., & Soller, A. (2005). *Cell Biology Education, 4*, 42–57.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement, 20*, 345-354.

Tatsuoka, K. K. (1995). Architecture of knowledge structures and cognitive diagnosis: A statistical pattern recognition and classification approach. In P. Nichols, S. Chipman, & R. Brennan (Eds.), *Cognitively Diagnostic Assessment* (pp. 327-359). Hillsdale, NJ: Erlbaum

Vendlinski, T. & Stevens, R. (2000, June). *The use of artificial neural nets (ANN) to help evalute student problem solving strategies*. Paper presented at the annual meeting of the international conference of the learning sciences. Mahwah, NJ.

Wason, P. C. (1966). Reasoning. In B. M. Foss (Ed.), *New Horizons in Psychology*. Penguin.

Williamson, D. M. & Bejar, I. I. (2000, April). *Kohonen self-organizing maps in validity maintenance for automated scoring of constructed response*. Paper presented at the annual meeting of the National Council on Measurement in Education. New Orleans, LA.

Williamson, D. M., Bejar, I. I. & Sax, A. (2004). Automated tools for subject matter expert evaluation of automated scoring. *Applied Measurement in Education, 17*, 323-357.

Chapter 5 - Experiment One: SOM Representation of Ordered Classes

The present research is devoted to understanding how well Self-Organizing Maps (SOMs) represent the test structure from simulated educational test data that comprise ordered classes. Simulated test results were chosen so that the ability of the SOM to recover known properties of data could be evaluated. Ordered classes are examined because they represent important characteristics of educational data. First, ordering of the simulated data reflects differences in student competency and enables the comparison of the ordered representation created by the SOM with the simulated differences in student competency. The extent to which the ordering in the SOM reflects the order of the data constitutes evidence for the appropriateness of SOMs in the educational measurement context. Second, the simulated data were composed of classes since a trained SOM defines a centroid for each point in the map. Each centroid may reflect defining characteristics of the examinees belonging to its corresponding cluster. If the characteristics revealed by the centroids in the SOM match the characteristics of the simulated classes, this would be evidence that the SOM can preserve class features of examinees' performance.

Evaluating the potential of the SOM in representing ordered classes also involves a consideration of the boundary conditions under which characteristics of the original data are preserved. If the SOM is capable of providing useful information only about data that are in a form not commonly encountered in educational measurement, (e.g., very high item discriminations), then the appropriateness of SOMs to these applications will be limited. Furthermore, characteristics of the SOM may play a critical role in the accurate representation of the underlying data. For example, faithful depiction of the data

by the SOM may only be possible when certain parameters of the map (e.g., map size) are appropriate for the data.

In the present experiment, three levels of item discrimination are crossed with two levels of SOM size to shed light on the specific conditions under which the SOM is appropriate for the analysis of educational data. Item discrimination is the strength of relationship between item level performance and the ability level of the examinee and is an index of item quality in educational measurement. This variable represents the amount of uncertainty or randomness in predicting item level performance from the ability of the examinee. This is an important variable because it determines how much noise the SOM will tolerate while still producing an accurate representation of the data. The size of the SOM will determine how specific the representation of the data will be, and whether this representation is true to the original data. Because each model vector will represent non-overlapping regions of the original data, the average number of data points represented by each model vector will decrease in a large map, resulting in an increased selectivity of each vector. As a result of this increased selectivity, extraneous characteristics of data may be prominent in the map at the expense of its overall structure. It is therefore an important question for the present experiment to shed light on the appropriate size of the map to accurately render the structure of the original data.

Method

Characteristics of the Data

A program written in Microsoft Excel in Microsoft Visual Basic for Applications (VBA) by the author generated all data (see Appendix A). Characteristics of the data for Experiment One are listed in Table 5.1. Four steps are required to generate the data sets,

(1) specifying the test structure, (2) specifying the distribution of ability in the simulated examinees, (3) specifying the distribution of item difficulty, and (4), determining the correctness of the responses. These steps are described below.

Test Structure

The first step in determining the responses of simulated examinees is to specify the test structure. The data in the first study consisted of 4 unidimensionally ordered classes. These defined the possible states of mastery to which a simulated examinee could be assigned as well as the states of mastery to which individual items were targeted. Each of these states was then defined in terms of the specific ability level of the simulated examinee and the difficulty level of the items. From these values, the probability of a correct response could be generated. For example, a given item might have a difficulty value of 1.0, and if the simulated examinee possesses an ability of 1.0 or greater, he is likely to answer this item correctly.

Examinee Characteristics

The second step in the simulation process is to specify the distributions of knowledge states within the population of simulated examinees. In each condition, examinees were first described in terms of the class to which each belonged and then the ability level that represented that state. Specifying these values enabled direct comparison of item and examinee characteristics in order to determine the correctness of a given response.

An equal number of examinees were simulated to be at one of n_k levels of ability, where n_k is the number of latent classes in the condition. The ability level (θ) of the examinee was determined by finding the z-score of the cumulative probability defined by

$(2k - 1) / 2n_k$, where k is the latent class. For the present study, 125 examinees were simulated for each class and their θ-values were defined as the z-score associated with a cumulative probability of 0.125, 0.375, 0.625, and 0.875. These probabilities corresponded to θ-values of –1.15, -.32, .32, and 1.15, respectively.

Item Characteristics

The third step in this simulation is to specify the distribution of item characteristics. In order to specify items that were 'targeted' to a particular latent class, the difficulty parameters of the items were set to be equal to the θ-values corresponding to each class. As shall be seen in the section describing how the responses are determined, this is equivalent to setting to 0.5 the probability that examinees belonging to a given class answered correctly items belonging to that class. Three items were targeted to each class.

Determining the Responses

The last step is to run the simulation, the end result of which is a complete data set representing correct and incorrect responses for each simulated examinee on each item. Whether or not a given item was answered correctly for a given simulated examinee depended on the comparison of the item and examinee characteristics. For the present study, this was carried out using the two-parameter logistic (2PL) IRT model. The equation for determining probability correct from the characteristics of items and examinees is:

$$P(u = 1|\theta) = \frac{1}{1 + e^{-1.7a(\theta - b)}}, \qquad (5.1)$$

where u =1 is a correct response, θ is examinee ability, b is the item difficulty parameter and a is the item discrimination parameter. Three values for item discrimination were chosen (2.0, 1.0, 0.5), and therefore three complete sets of data were generated for the present experiment. In order to determine whether the response to the item was scored as correct, a random number between zero and one was generated and compared to the probability calculated from the above formula. If the value of the random number was equal to or lower than the number calculated from the formula, the item was scored as correct and incorrect otherwise. This last step in the procedure was carried out for each simulated examinee and for each item until each complete data set was generated.

Training the SOM

As with the data generation program, the author created the SOM program in Microsoft Excel using Microsoft Visual Basic for Applications (see Appendix B).

Network Architecture

Two sizes of SOMs were used in the first study (4x4 [16 units], 8x8 [64 units]). The number of input units corresponded to the number of test items, in this case 12. The training sets were one of the 3 sets of 500 patterns generated using the parameters mentioned in the previous section.

Training Method

Recall from Chapter 4 that each of the output units has a model vector whose number of elements is equal to the number of units in the input layer. Training proceeded stepwise as follows. First, the difference between the current input pattern and each model vector was calculated to determine the unit with the smallest root-mean squared error for the pattern. This is equivalent to choosing the model vector closest in Euclidian

space to the input pattern. Model vectors for all units within a given radius (i.e., the neighbourhood) of the winning unit were then changed to lessen the difference between each element of these model vectors and the corresponding element in the current input pattern. Vectors were changed according to:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \ , \tag{5.2}$$

where m_i is a model vector for units within the neighbourhood of the winning unit, x(t) is the current input pattern, t is time, and $\alpha(t)$ is the value of the learning rate parameter. Model vectors for all other units in the SOM remained unchanged. The procedure by which a value for α is set is discussed below.

Training patterns were presented in a random order without replacement for 10000 or 40000 iterations for small (4x4) and large (8x8) maps, respectively. Following Kohonen (2001, p. 114), the starting radius of the neighbourhood function was defined as one half of the radius of the map, which in the present case was calculated as one half the average of the length and width of the map, that is:

$$\tfrac{1}{2}(X_{SOM} + Y_{SOM})/2 \tag{5.3}$$

where X_{SOM} is the width of SOM and Y_{SOM} is the length of SOM. Following Kohonen (1990), the learning rate α was changed in two phases: the ordering phase (iteration $\leq$ 2000) and the fine-tuning phase (iteration $>$ 2000). During the ordering phase, α

decreased linearly from 1 to 0.04 and during the fine-tuning phase, α decreased linearly from 0.04 to 0.005. Last, 100 replications of the simulation are conducted to determine the stability of the SOM solution (e.g., deBodt, Cottrell, & Verleysen, 2002).

Table 5.1. Characteristics of the Training, Data, and the Self-Organizing Map for Experiment One.

Variables	Values for Variables
Number of Latent Classes	4
Number of Items per Class	3
Number of Examinees	
Class 1 (θ = -1.15)	125
Class 2 (θ = -0.32)	125
Class 3 (θ = 0.32)	125
Class 4 (θ = 1.15)	125
Dimensions of SOM –	
Conds. 1-3 (Height, Width)	4, 4
Dimensions of SOM –	
Conds. 4-6 (Height, Width)	8, 8
Number of Iterations	
Conds. 1-3	10000
Number of Iterations	
Conds. 4-6	40000
Iterations Before Update	1

Number of Replications	100
Simulated Items Probabilities	
Conditions 1, 4	
Item Discrimination = 2.0	
Class 1 (θ = -1.15)	0.50, 0.50, 0.50, 0.06, 0.06, 0.06, 0.01, 0.01, 0.01, 0.00, 0.00, 0.00
Class 2 (θ = -0.32)	0.94, 0.94, 0.94, 0.50, 0.50, 0.50, 0.06, 0.06, 0.06, 0.01, 0.01, 0.01
Class 3 (θ = 0.32)	0.99, 0.99, 0.99, 0.94, 0.94, 0.94, 0.50, 0.50, 0.50, 0.06, 0.06, 0.06
Class 4 (θ = 1.15)	1.00, 1.00, 1.00, 0.99, 0.99, 0.99, 0.94, 0.94, 0.94, 0.50, 0.50, 0.50
Conditions 2, 5	
Item Discrimination = 1.0	
Class 1 (θ = -1.15)	0.50, 0.50, 0.50, 0.20, 0.20, 0.20, 0.08, 0.08, 0.08, 0.02, 0.02, 0.02
Class 2 (θ = -0.32)	0.80, 0.80, 0.80, 0.50, 0.50, 0.50, 0.20, 0.20, 0.20, 0.08, 0.08, 0.08
Class 3 (θ = 0.32)	0.92, 0.92, 0.92, 0.80, 0.80, 0.80, 0.50, 0.50, 0.50, 0.20, 0.20, 0.20
Class 4 (θ = 1.15)	0.98, 0.98, 0.98, 0.92, 0.92, 0.92, 0.80, 0.80, 0.80, 0.50, 0.50, 0.50
Conditions 3, 6	
Item Discrimination = 0.5	
Class 1 (θ = -1.15)	0.50, 0.50, 0.50, 0.33, 0.33, 0.33, 0.22, 0.22, 0.22, 0.12, 0.12, 0.12
Class 2 (θ = -0.32)	0.67, 0.67, 0.67, 0.50, 0.50, 0.50, 0.33, 0.33, 0.33, 0.22, 0.22, 0.22
Class 3 (θ = 0.32)	0.78, 0.78, 0.78, 0.67, 0.67, 0.67, 0.50, 0.50, 0.50, 0.33, 0.33, 0.33
Class 4 (θ = 1.15)	0.88, 0.88, 0.88, 0.78, 0.78, 0.78, 0.67, 0.67, 0.67, 0.50, 0.50, 0.50

Analysis of SOM Performance

Two main questions guided the analysis of the SOM performance:

1) What can Self-Organizing Maps reveal about the essential characteristics of test data comprising ordered classes?
2) What are the conditions under which Self-Organizing Maps succeed and fail to reveal these characteristics?

Further, each of the above issues is examined from three perspectives:

a) statistical, in which key indexes of the SOM representation of the input data will be analyzed across conditions,
b) qualitative, where SOM's derived from specific replications will be graphically inspected for the faithfulness of their representation of the ordered classes, and
c) interpretive, in which the characteristics of model vectors are examined to better understand the representational capabilities of the SOM for educational measurement.

The results are organized around these three perspectives, as follows. In the first section, statistical analyses are conducted on three measures, quantization error (QE), topological preservation (TP), and correlation among distances (R_{dist}). QE reveals the extent to which vectors in the input data are different from their corresponding representation in the SOM. TP and R_{dist} show how well spatial interrelationships between points in the original data are preserved in the SOM. The second section focuses on the qualitative perspective and consequently, replications from each condition are visually examined in order to shed light on how class information is preserved in the SOMs. The

last section uses multi-dimensional scaling to interpret the dimensions of the SOM in terms of item and test variables. Taken together, these analyses provide an examination of the SOMs in representing the structure of simulated test data.

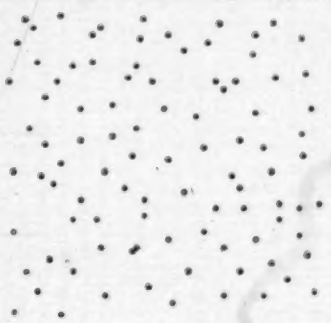
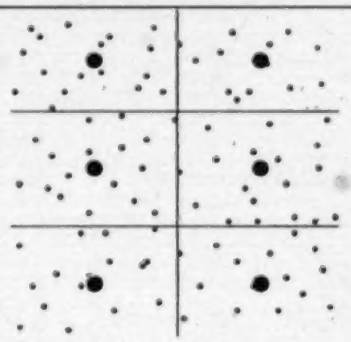
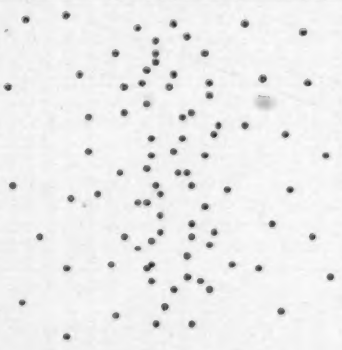
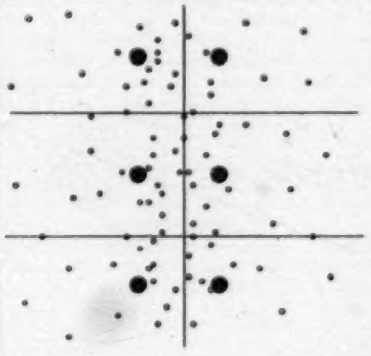
Map-Data Fit: Quantization Error

One of the characteristics of a SOM is that its organization reflects the probability density of the input data. This characteristic is reflected in SOM organization in two ways: a) the location of the model vectors, and b) the concentration of each of these vectors in space of the input data. Consider the data presented in Figure 5.1. In the left hand column are two scatterplots of two-dimensional data that could be analyzed by a SOM. The first of these scatterplots has approximately uniform probability density across both the X and Y dimensions while the second has an approximately normal probability density across the X dimension and uniform across the Y dimension. Both scatterplots have an identical number of points. The right hand column displays these same data with model vectors and receptive fields from a hypothetical analysis with a 2x3 SOM. Recall that the receptive fields designate the regions in which all points are closest to a single model vector. In the figure, the dark circles represent the location of the model vectors after analysis and the lines represent the boundaries of the receptive field for each model vector.

Note the locations of the model vectors in each of the two scatterplots. For the top one, the model vectors, like the data, are approximately uniformly distributed across the surface demarcated by the data. In the second, these vectors are uniform across the Y dimension, but are more centrally concentrated in the X dimension also reflecting the density of the underlying data. This organization, too, reflects characteristics of the

underlying data. This simple demonstration shows how the placement and concentration of these model vectors can represent characteristics of the probability density of the data in each of the scatterplots. How does the SOM come to place model vectors in ways that reflect this density? And how can the faithfulness of the representation of probability density be evaluated objectively?

Figure 5.1. SOM Representation of Probability Density.

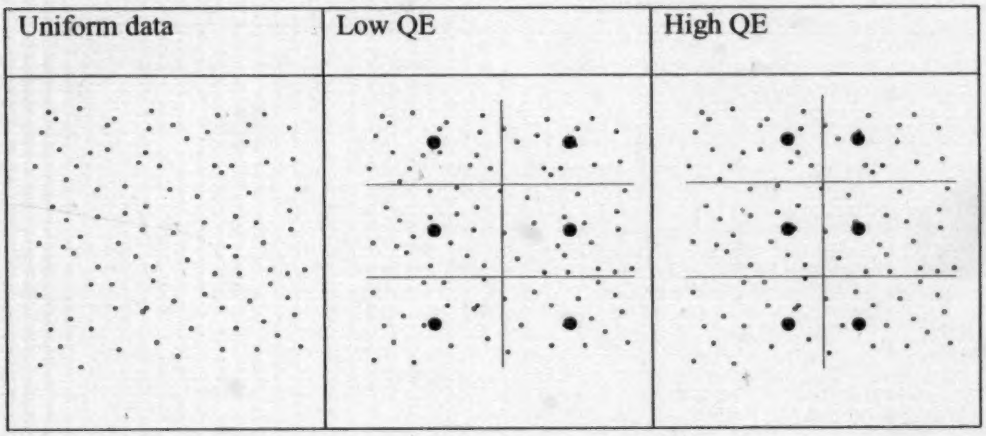
Probability Density	Scatterplot	Scatterplot with Model Vectors and Receptive Fields from a hypothetical SOM
X - Uniform Y – Uniform		
X – Normal Y – Uniform		

Recall that the locations of the model vectors are the end product of training the SOM. The goal of this training is to minimize the average Euclidian distance (equation 4.1) between each model vector and each of the data vectors in its receptive field. Since this average distance defines how successfully a network is trained, it is a critical measure of the overall quality of the SOM. This measure is referred to as *Quantization Error* since it defines the expected amount of error when the model vector is used to 'stand in' for individual data; that is, when a data vector is quantized to the nearest model vector. But how does a well-trained map (i.e., one for which QE is a minimum) for a given set of data faithfully reflect the probability density of those data? To see this, consider that a heuristic for minimizing QE is to place each model vector in a location where in its own receptive field, there are many data vectors close to it and few that are far away. Returning to Figure 5.1, it can be clearly seen that the model vectors in the figure adhere to this heuristic; they are close to as many such vectors as possible, and far away from few. Since it has been noted that the placement of the model vectors in the figure does reflect the probability density of the data, this is suggestive of a connection between minimizing QE and faithfully representing probability density.

The existence of this connection implies that examining QE for sets of model vectors, that is, those derived from several 'runs' of a SOM, would be effective in comparing the map-data fit of these runs. Suppose that the model vectors in the middle and right-hand panel of Figure 5.2 were presented as candidate SOM representations of the data in the left-hand panel. It can be determined using the above heuristic that the vectors from the right-hand figure do not represent well data of uniform probability density; as compared with those from the middle panel, more data are farther away from

the model vectors (i.e., those on the extreme right- and left-hand sides of the panel) and fewer are close within each model vector's receptive field. QE in this scenario would be non-optimal, and greater than the QE derived from the model vectors displayed in the middle panel. The importance of this comparison for the present research is that it will help determine the run with the best map-data fit.

Figure 5.2. The Position of Model Vectors and Resulting Quantization Error.



Thus far, the meaning of differences in QE for SOMs derived from the same set of data has been a point of focus. Consider the situation where identical SOMs are trained on two different sets of data. What would differences in QE reveal about the structure of each respective set of data? Assuming that QE is a minimum for each SOM, smaller QE implies that the data comprise regions of greater and lesser probability density. To see why this is so, consider again the data presented in Figure 5.1. The data in the top part of the figure have essentially uniform probability. In contrast, the data in the bottom part of the figure shows regions of high and low probability density and thus is more

differentiated. Using the previous heuristic, it can be surmised that this more differentiated data will have lower QE; as compared with the scatterplot in top part of the figure, more data are close to the model vectors in the bottom part of the figure and fewer are far away. This characteristic will also be important for the present research in understanding how data with different characteristics are represented by a SOM.

In summary, an important characteristic of a SOM is to represent the probability density of data. A key measure of the success of the map in representing this probability density is Quantization Error. Using a simple heuristic, it has been shown that, given the same data and SOM, low QE reflects the placement of model vectors where they are closest to the most data, that is, in regions of high probability density. Thus, comparing QE from several runs of a SOM is an effective way of determining which of the runs best represents the probability density of the data. Furthermore, comparing the QE from the same SOM of two different sets of data can reveal which data set is more differentiated with respect to probability density.

Projection

The effectiveness of the SOM to maintain order relationships that are present in the test data is reflected in several measures of projection. Since the simulated test data are scored responses to 12 hypothetical test items, points representing these data could be considered to populate a 12-dimensional space. This larger space containing the scored responses will be referred to as the *metric* space. Because the SOM is typically two-dimensional, significant reduction in the complexity of the data in the metric space is likely to occur for it to be represented in the map. Maintaining relationships in the mapping of points from the metric space to the SOM means that despite the reduction in

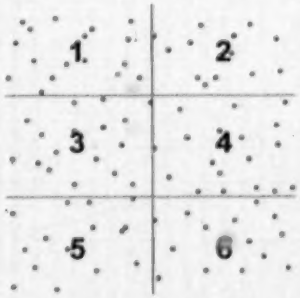
complexity, the map is representing well the 'dominant' dimensions in the test data. High values on measures of projection imply that relationships among points are not lost when data are projected from the high- to low-dimensional space.

Two facets of the relationship among points are the focus of measures of projection, *adjacency* and *distance*. The preservation of adjacency relationships implies that when co-ordinates in the SOM are adjacent, their respective receptive fields in the metric space ought to be adjacent also. The preservation of distance holds that the representation of distance between points in the SOM is a good approximation of the distances between the model vectors, and thus the points that comprise them. This implies that there ought to be a strong positive correlation between distances between co-ordinates in the SOM and the corresponding distances between the model vectors they represent.

In Figure 5.1, the model vectors and the points in the SOM are not differentiated; they are presented as the same. Of course, model vectors populate the metric space and points in the SOM populate a co-ordinate space. It is the relationship between these two spaces and the points that populate them that is fundamental to the notion of projection. In order to see more clearly the relationship between the two spaces, Figure 5.3 presents model vectors and SOM co-ordinates as disengaged. In the left-hand panel, model vectors presented as numbers are shown as populating the metric space. In the middle panel, co-ordinate points representing each model vector are shown on a SOM. Note that the location of these co-ordinate points is appropriate; their relative placement in the SOM reflect the relative placements of the model vectors in the metric space. More concretely, the points that are adjacent in the SOM correspond to model vectors whose

receptive fields are adjacent in the metric space. Also, distance between points in the SOM is a good analogy for distance between their corresponding model vectors.

Figure 5.3. Projection Quality of Model Vectors in the Metric Space as a Function of the Placement of Corresponding Co-ordinates in the Self-Organizing Map

Model Vectors in Metric Space	Co-ordinates in SOM: Well-Projected		Co-ordinates in SOM: Poorly Projected	
	1	2	6	2
	3	4	3	4
	5	6	5	1

In contrast, the location of the co-ordinate points in the SOM shown in the right-hand panel is a poor representation of the data in the metric space. Fewer points that are adjacent in the SOM have corresponding model vectors whose receptive fields are adjacent in the metric space. For example, in the SOM, point 6 is adjacent to point 2, whereas the receptive field corresponding to model vector 6 in the metric space is not adjacent to the receptive field belonging to model vector 2. Furthermore, distances between points in the right-hand panel do not represent well the distances between corresponding points in the metric space. For example, the ordering of model vectors in increasing distance from model vector 1 is 2 and 3 (equal), 4, 5, and 6. In the poorly-fitting SOM, this same ordering of distances is 4 and 5 (equal), 3, 2, and 6. Note that for

each SOM in Figure 5.3, the QE could be the same; the expected, or average distance between each data point and its respective model vector in the metric space is not affected by the how the co-ordinates in the SOM are arranged. Said another way, QE relates only to the metric space, not the SOM space. However, since the model vectors *represented* by each location in the map have changed, the faithfulness of the SOM's projection of those data is affected.

As mentioned above, measures that quantify projection are focused on two facets of the relationship among points between the metric space and co-ordinates in the SOM, adjacency and distance. In first measure, following Villman, Der, Herrman, and Martinetz, (1997), the two closest model vectors to each data point are determined and their corresponding co-ordinates in the SOM are examined for their adjacency. The proportion of all points in the data for which the two closest model vectors map to adjacent points in the SOM defines a measure called *Topological Preservation* (TP). Applying this method to the SOM in the right-hand panel in Figure 5.3 it can be seen that a significant proportion of points in the metric space will have model vectors with adjacent receptive fields, but not adjacent points in the corresponding SOM (e.g., those for whom model vectors 5 and 6 are the two closest). All such points will highlight the inaccurate projection of the input data on the SOM and consequently will contribute to a diminished value of TP.

Maintaining distance relationships is evaluated by the correlation of distances between analogous points in the two spaces. For this measure referred to here as R_{dist}, a matrix of distances is created between co-ordinates in the SOM and between all the model vectors in the input data. Each distance in the SOM is matched to its analogous

distance in the metric space by matching the points of the SOM to their respective model vectors. A correlation analysis is then performed on all corresponding distances. If the correlation is large and positive, it can be concluded that distances in the map are a good representation of distances in the input data. In other words, the input data are projected well on the SOM. Noting that correlation between sets of data is closely related to the similarity in ordering between those data, the example above describing the poor ordering between distances between model vectors and corresponding distances between SOM co-ordinates highlights the effectiveness of this measure for projection.

In the present research, the preservation of adjacency and distances reveal not only how well the SOM represents the original data, but also how easily the data can be rendered in a lower-dimensional space. These two characteristics, the ability of the SOM to represent data and the characteristics of the data that allow them to be well represented are evaluated below.

Results

Section I – Statistical Results

Quantization Error

In order to see the effect of the various conditions in Experiment 1 on QE, a 2 x 3 (SOM Size x Item Discrimination) analysis of variance (ANOVA) was conducted. The interaction between these factors was highly statistically significant, $F(2, 594) = 637.3$, $p < 0.001$, indicating that the mean values of QE were not predicted by each main effect alone. However, the large differences between means in relation to the small standard errors show that the significant interaction mainly reflects only small absolute deviations from the strong main effects and results from a large amount of statistical power. In this

context, post hoc tests are difficult to interpret and therefore are not provided. The two main effects were also significant, for SOM size, $F(1, 594) = 117616.2$, $p<0.001$, and for Item Discrimination, $F(2, 594) = 141224.8$, $p<0.001$. QE was consistently smaller for larger maps implying that a larger number of points provide a higher resolution of the original data. In addition, QE consistently decreased as item discrimination increased which indicates that the SOM can more precisely fit model vectors to each point in their receptive fields when the class distributions are more discrete and thus the data are more differentiated with respect to probability density.

Table 5.2. Mean (Standard Error) Quantization Error by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
4 x 4	1.21 (0.000)	0.95 (0.001)	0.63 (0.002)	0.93 (0.014)
8 x 8	0.92 (0.001)	0.63 (0.001)	0.25 (0.002)	0.60 (0.016)
All	1.06 (0.010)	0.79 (0.011)	0.44 (0.013)	0.76 (0.012)

Projection I: Topological preservation

An ANOVA comprising the same factors as the analysis of QE was conducted on TP, the measure reflecting the preservation of adjacency relationships. The Item Discrimination by Map Size interaction was statistically significant, $F(2, 594) = 34.0$, $p<0.001$, indicating that differences in TP due to Item Discrimination depended on the size of the SOM. An examination of the means presented in Table 5.2 suggests that this significant interaction resulted from a floor effect for TP in the large map and the Item

Discriminations of 0.5 and 1.0. As in the analysis of QE, the specific comparisons implied by this interpretation are difficult to test statistically because the small standard errors and large sample size imply a large amount of statistical power. The main effects for both Item Discrimination and Map Size were statistically significant, $F(2, 594) = 447.1$, $p<0.001$ and $F(1, 594) = 13231.8$, $p<0.01$, respectively. These results show that higher values of Item Discrimination have a positive impact on projection, while large maps have a negative effect.

Table 5.3. Mean (Standard Error) Topological Preservation by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
4 x 4	0.65 (0.004)	0.75 (0.004)	0.83 (0.005)	0.74 (0.005)
8 x 8	0.24 (0.003)	0.27 (0.005)	0.35 (0.008)	0.29 (0.004)
All	0.45 (0.015)	0.51 (0.017)	0.59 (0.018)	0.51 (0.010)

Projection II: Correlations of distances

A 2 x 3 (Map Size x Item Discrimination) ANOVA was conducted on the Correlation of Distances to determine how it was impacted by the different conditions. In all analyses involving R_{dist}, Fisher's (1915) z-transformation of the correlations was first performed to create a linear scale more appropriate both to creating averages and performing the ANOVA. All means reported are transformed back into the correlation metric. A statistically significant interaction on the transformed values was found, $F(2, 594) = 67.9$, $p<0.001$ and appeared to result from a ceiling effect across Item

Discrimination values of 1.0 and 2.0 affecting both conditions of map size (see Table 5.3). Like the two previous analyses, this interaction is difficult to confirm statistically because of the small standard errors and large amount of statistical power involved. The results from the two main effects on R_{dist} reflect the same pattern as seen with TP. The main effect of Item Discrimination was statistical significant, $F(2, 594) = 666.7$, $p<0.001$, and shows that the positive impact on projection of higher discriminations. There was a statistically significant main effect of Map Size ($F[1, 594] = 6220.4$, $p<0.001$), revealing that larger maps have lower values of R_{dist}.

Table 5.4. Mean (Standard Error) Correlations (R_{dist}) between Distances in Metric Space and Associated SOM Co-ordinates by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
4 x 4	0.71 (0.003)	0.76 (0.001)	0.75 (0.002)	0.74 (0.002)
8 x 8	0.53 (0.002)	0.63 (0.002)	0.64 (0.002)	0.60 (0.003)
All	0.63 (0.007)	0.70 (0.005)	0.70 (0.004)	0.68 (0.003)

Discussion

It is clear from the above results that both SOM Size and Item Discrimination have significant impact on the representation of data structure. In general, the representation was more accurate with higher values of Item Discrimination resulting in lower values of QE and higher values for both TP and R_{dist}. This was not the case for SOM Size, however, as more units resulted in less QE, but lower values for the measures

of projection. This may have occurred since a larger number of units imply that there is a lower probability that any two units will be adjacent. A qualitative examination of the SOMs in each condition presented in the following section will shed light on this finding.

It is unclear how to interpret the absolute values of QE but their consistency within conditions as revealed by the small standard errors in Table 5.2 suggests that they are approaching a minimum. The magnitude of TP and R_{dist} reveal that the projection capabilities of the SOM for unidimensionally ordered classes in this research are somewhat limited. The preservation of adjacency relationships as indexed by the measure of TP appears to be reasonable for the best condition (SOM Size = 4 x 4, Item Discrimination = 2.0), with an average of 83% preserved. However, the mean correlation between distances in the original data and the SOM in this same condition was 0.75, or on average, just over half of the variance accounted for. As would be expected, equal or lower values on these measures were observed for all other conditions. One explanation for these small values is the difficulty of a two-dimensional SOM representing essentially one-dimensional data. This possibility is discussed in a following section.

Section II – Qualitative Examination of the SOM

In examining the model vectors from each replication, an important characteristic of the SOMs representing dichotomous data is revealed. This characteristic is that the model vectors comprise the unit conditional item probabilities. That is, the value of each element of a model vector converges to the proportion of correct responses to the associated item for the simulated examinees classified to the given unit. To understand why this is so consider that before the end of training, the neighbourhood function decreases in size to include only a single unit. Under this condition, each model vector

becomes a centroid, or multidimensional average for the set of data it represents. Since the test data are dichotomous, this average is simply the number of correct responses divided by the number of respondents, i.e., the proportion of correct responses.

To see this result more clearly, consider the two model vectors presented in Table 5.5. The first model vector represents performance of high achieving examinees. This follows from the observation that elements of the vectors are the probabilities of answering each item correctly. Viewed in this light, it can be seen that each examinee classified to this unit had correct responses for items 1 through 5, and for number 12. No examinee had a correct response for item 10. All other items were correctly answered according to the probabilities listed. The second model vector represents low achieving examinees. These examinees responded incorrectly to items 4, 5 and 8 through 12, but had correct responses for items 2 and 6. Furthermore, summing all item probabilities together provides the average total score achieved by all examinees classified to the unit. These average scores are 10.25 and 3.71 out of twelve for examinees classified to units 1 and 2 respectively. This result, that model vectors comprise unit conditional item probabilities, provides important information for the interpretation of the model vectors themselves, and the interpretation of the whole SOM.

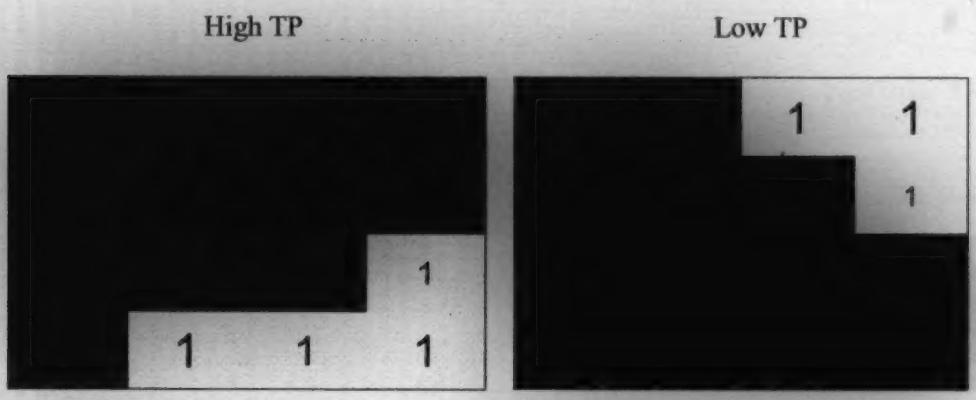
Table 5.5. Two Sample Model Vectors

	Item											
Model Vector	1	2	3	4	5	6	7	8	9	10	11	12
1	1.00	1.00	1.00	1.00	1.00	0.96	0.91	0.88	0.96	0.00	0.54	1.00
2	0.70	1.00	0.94	0.00	0.00	1.00	0.06	0.00	0.00	0.00	0.00	0.00

Extremes of TP

By visually inspecting the SOM solutions, a better sense of how the SOM is organizing the data can be gained. For this purpose, the replications with the highest and lowest values of TP are displayed in Figure 5.2. The numbers in the figure denote the class that is most often classified to each unit in the map. The size of the number is proportional to the number of observations classified to that point in the map.

Figure 5.2. Most Frequent Intended Class Membership for Two Self-Organizing Maps in Experiment One (Condition = Small Map, Item Discrimination 2.0).



Several features of these maps are noteworthy. First, all units representing a single class tend to be grouped together. When adjacent units represent different classes, in most instances the class numbers differ only by one, reflecting a smooth ordering of the model vectors. There are no instances where units representing Classes 1 and 4 are adjacent. Where adjacent units do not represent consecutive classes, the number of observations tends to be small. Last, there is little to differentiate highest and lowest values of TP for Condition 1 by visually examining the maps.

Typical Maps for each Condition

The effect of Item Discrimination on the structure of the maps is examined in Figures 5.3 and 5.4. Figure 5.3 displays 'typical' maps for each of the small map conditions. Figure 5.4 displays these maps for the large map conditions. A typical map was determined by selecting randomly from those replications whose values of QE, TP, and R_{dist} were within one standard deviation of the mean. The values for Item Discrimination for the maps in Figure 5.3 and 5.4 are, from left to right, 2.0, 1.0, and 0.5. For the typical 4 x 4 maps, the same characteristics as mentioned for the maps of highest and lowest TP (Figure 5.2) also apply. Namely, regions representing the same class are contiguous; the large majority of adjacent units that are not the same class have class numbers that differ by one while those that differ by more than one have small numbers of members. Again, there is little in the visual structure of the map to differentiate between conditions. This is significant. The data that were used to generate these maps differed considerably in their discrimination. This suggests that the quality of visual interpretation of data from unidimensionally ordered classes is robust against variation in item discrimination, at least for the values in these conditions.

Despite this robustness, there must be differences in how homogeneous the individual units in the map are with respect to the classes of their members. This is true because as item discrimination decreases, more overlap is introduced between the class distributions. One way to determine the homogeneity of class membership is to determine how accurate classification performance is if it is assumed that all members of a particular unit are categorized as belonging to the most frequent intended class. The

percentage of matches for the entire map is the average percentage that the members of a unit belong to the modal class.

Figure 5.3. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment One, Small Map Conditions

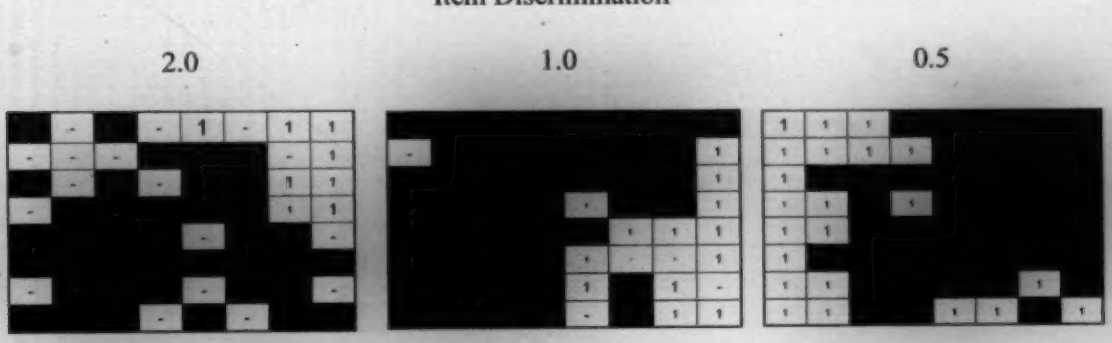
To examine this homogeneity, average classification performance was examined across each the three conditions represented in the above figure. The means and standard deviations for these conditions (maximum 500) are presented in Table 5.6. For the conditions representing high, medium, and low Item Discrimination, the respective means were 403.6 (80.1%), 316.9 (63.4%), and 257.2 (51.4%). For the replications depicted in Figure 5.3, the frequency of matches was 397, 319, and 255, very close to the mean values. Thus, it appears that despite wide variation in the homogeneity of the underlying classes in the present experiment, the 4x4 SOM projects data comprising unidimensionally ordered classes similarly.

Table 5.6. Mean (Standard Deviation) Number of Matches between Intended Class and Modal Class by Condition (Maximum = 500)

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
4 x 4	257.16 (6.42)	316.90 (4.90)	403.57 (6.95)	325.88 (60.52)
8 x 8	292.81 (5.66)	346.30 (4.80)	422.69 (3.12)	353.93 (53.59)
All	274.99 (18.86)	331.60 (15.51)	413.13 (10.99)	339.91 (58.81)

For the larger maps, this variation in homogeneity does affect the visual interpretation of the SOM. Figure 5.4 displays typical maps for Conditions 4, 5, and 6. The characteristics of the typical maps from Conditions 1, 2, and 3, namely, the contiguity of regions representing the same class and the overwhelming tendency for adjacent units to be of identical or differing by one class holds for the typical map from Condition 5. The Condition 4 map fails to display these characteristics only because of the significant number of units to which no observations were classified. If it were decided that these 'vacant' units were members of the classes to which they are adjacent, the properties of other typical maps would be manifest. Interestingly, these vacant units also must at least partially account for the low values of TP for Condition 4; fewer possibilities for adjacency relationships can exist for units adjacent to vacant units.

Figure 5.4. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment One, Large Map Conditions



Like Condition 4, the map representing a typical replication from Condition 6 fails to display the characteristics of maps from typical replications of Conditions 1 through 3. In this case, the lack of contiguity and the increased prevalence of non-identical and non-consecutive class units appears to reflect the increased overlap of the class distributions comprising items of low discrimination. Why would this difference be seen in only the large map condition? One explanation is that although the ordering of units in this map is proper in that the Euclidian distances between model vectors are smaller for closer units, the degree of class overlap is sufficiently extensive that the modal intended class is not a reliable measure of the likely class for individual units. The proportion of each intended class at each unit was intended as a proxy for population class conditional probability. However, when the map becomes so large that the number of observations classified to some units become very small, the possibility increases that this proportion does not accurately reflect the population class conditional probability. When this inaccuracy is significant enough to change the modal intended class, misclassification of a unit will result.

Section III – Interpretation of the SOM

Dimensions of the SOM

From visual inspection of typical maps it can be seen that in most conditions, the SOM orders model vectors according to their most probable intended class. This is important because it demonstrates that the SOM reveals the key characteristic of the simulated data. However, as encouraging as this ordering is, it is at odds with the low values for measures of projection. How might these two seemingly disparate findings be reconciled? One way that this issue might be addressed is to note that although visual inspection of the typical replications reveals an ordering of classes, this ordering is manifest in only one direction of the map. Since measures of projection will take into account both dimensions, knowing on what basis the SOM orders model vectors in the direction orthogonal to the intended class could illuminate the low values for projection.

To address this issue, it is essential to be able to determine what characteristics of examinee responses each unit represents. When this is known, each dimension of the map may be describable in these terms. One way to approach this is to project the model vectors into a metric space using multidimensional scaling (MDS). Analyzing each of the dimensions revealed by MDS may provide insight into the basis by which the SOM is organized.

Each of the typical replications from above was analyzed using ALSCAL (Takane, Young, & deLeeuw, 1977). Since the primary interest is the projection of the model vectors in two dimensions, MDS analyses are reported for one and two dimensions only. The values of stress and the variance accounted for (VAF) in each replication for one and two dimensions are listed in Table 5.7. Two trends emerge from these results. First, as

item discrimination decreases, values of stress increase and the dispersion accounted for decreases. This is predictable, as decreases in item discrimination is synonymous with increases in random error. Second, for the same number of dimensions in the MDS model, larger maps are more difficult to fit, as evidenced by the larger values of stress and lower dispersion accounted for as compared with the smaller maps.

Table 5.7. Stress1 and Variance Accounted For (VAF) in One- and Two-Dimensional MDS Analyses on Typical Replications in Each Condition

	MDS Analysis			
	One-Dimensional		Two-Dimensional	
Condition	Stress1	VAF	Stress1	VAF
1 (4x4, a = 2.0)	0.130	0.944	0.088	0.968
2 (4x4, a = 1.0)	0.205	0.862	0.119	0.928
3 (4x4, a = 0.5)	0.333	0.655	0.189	0.797
4 (8x8, a = 2.0)	0.214	0.865	0.130	0.931
5 (8x8, a = 1.0)	0.289	0.750	0.199	0.814
6 (8x8, a = 0.5)	0.440	0.427	0.292	0.535

Table 5.8 displays the correlations between the total expected score associated with the unit and the co-ordinates for each dimension and for each unit in each SOM. In all conditions and for both the one- and two-dimensional analyses, the first dimension co-ordinates were highly correlated with total score. This implies that the distances between model vectors are well predicted by the sum of their individual elements. This result

follows directly from the essential unidimensionality of the simulated data. Since two dimensions in the MDS projection are orthogonal to each other, the second dimension must be sensitive to variance not captured by total score. In other words, the Dimension Two must be sensitive to item performance not predicted by the examinee's total score.

Table 5.8. Correlation between Dimensional Co-ordinates of Each SOM Unit in MDS Analyses and Total Expected Score for Unit Members, Performed Separately for Typical Replications of Each Condition

	MDS Analysis		
	One-Dimensional	Two-Dimensional	
Condition	Dim 1	Dim 1	Dim 2
1 (4x4, a = 2.0)	0.999	-0.997	-0.190
2 (4x4, a = 1.0)	0.999	-0.998	-0.069
3 (4x4, a = 0.5)	0.988	-0.996	0.043
4 (8x8, a = 2.0)	0.998	-0.999	0.237
5 (8x8, a = 1.0)	0.995	0.996	-0.007
6 (8x8, a = 0.5)	-0.967	-0.988	0.124

To determine which items and which model vectors were flagging these deviations from unidimensionality, the co-ordinates from the second dimension of the MDS analysis were correlated with the unit conditional item probabilities for each model vector in the first ten replications of Condition One. Table 5.9 displays these correlations by item. Overwhelmingly, these correlations fail to meet statistical significance at the $p<0.05$

level. Since by chance alone, 6 out of 120 correlations are expected to be statistically significant and 3 were observed, there is no indication of a relationship between item-level performance and the second dimension from the MDS analysis. Because the data were simulated only according to a unidimensional model, this inconsistency is not surprising. However, the appropriateness of using a multidimensional model (i.e., the two-dimensional SOM) to represent these unidimensional data given this inconsistency is questionable.

Table 5.9. Correlation between Dimensional Co-ordinates and Unit Conditional Item Probabilities from Ten Replications of Condition One.

	Item Number											
Replication	1	2	3	4	5	6	7	8	9	10	11	12
1	-0.15	-0.48	-0.34	-0.42	-0.14	-0.07	0.10	0.04	0.20	0.48	0.24	0.27
2	-0.36	-0.25	0.15	0.05	-0.18	-0.41	-0.42	0.17	0.28	0.49	0.37	0.08
3	-0.02	-0.41	-0.22	0.10	-0.12	-0.45	-0.24	0.14	0.48	0.43	0.46	0.23
4	-0.44	-0.24	-0.18	-0.03	-0.26	-0.23	0.40	0.26	-0.11	0.49	0.35	0.23
5	-0.21	-0.31	-0.44	-0.47	0.07	-0.05	-0.03	0.47	0.13	0.21	0.21	-0.12
6	-0.26	-0.17	-0.33	0.03	-0.15	-0.47	0.16	0.06	0.25	0.31	0.18	0.38
7	0.01	-0.32	-0.19	-0.22	-0.08	-0.34	[redacted]	0.06	-0.01	0.12	-0.02	0.32
8	-0.08	-0.18	-0.19	-0.14	-0.17	[redacted]	0.02	0.18	0.18	0.44	0.29	0.32
9	-0.26	-0.42	-0.16	-0.39	0.07	-0.20	-0.32	0.12	0.25	0.40	0.46	0.44
10	0.06	-0.07	-0.12	0.05	-0.14	[redacted]	-0.36	0.25	0.27	0.39	0.43	0.19

Note: Highlighted values indicate statistical significance at the $p<0.05$ level.

General Discussion

According to the results of Experiment One, the usefulness of the two-dimensional SOM to represent unidimensionally ordered classes is unclear. On the positive side, there is evidence that across replications, the SOM finds a set of model vectors that approach a minimum error. This is evidenced by only small variations in Quantization Error across replications within the same condition, despite random starting values for the model vectors. With respect to projection, a visual inspection of typical replications within each condition demonstrates that the abilities of the simulated examinees are being represented in the placement of the model vectors. In particular, by using the modal intended class as a means of assigning classes to locations in the map, it can be seen that the SOM creates an ordering of model vectors that to some extent mirrors the ordering of examinee ability. In addition, this ordering was robust for the typical replications in each small map condition and suggests that under certain circumstances, the SOM can tolerate a considerable amount of random error to reproduce the essential characteristics of the data. Last, the result that the model vectors are the unit conditional item probabilities is intriguing. By examining the elements of the vector, an interpretation of the characteristics of the examinee responses classified to the unit can be made, both at the item and total score levels. Furthermore, this summary of examinee performance can be used to optimally classify new respondents to units in the map. That is, the likelihood of an examinee "belonging" to each unit can be evaluated by Bayesian classification. The maximum value of this likelihood given the examinee's responses determines the most probable unit for the examinee.

On the negative side, the capacity of the SOM to preserve properties of the original data in Experiment One is limited. For most conditions, and particularly in the large maps, topological preservation was low, which indicates that the basis for adjacency in the original space was not well identified by the SOM. Several factors could account for this. First, for the large maps, the number of units implies that any two units are less likely to be adjacent. Furthermore, the mean distance between adjacent units in the large map is less than that for the smaller map. This in turn implies a stricter criterion for adjacency in the large versus small maps; two units the same distance apart may be adjacent in the small map but not adjacent in the large map. Second, concerning the correlation of distances measure, recall that the model vectors in SOM are located only at whole number co-ordinates. Given this, the correlation of distances measure will be maximized only when adjacent model vectors in the metric space are similarly spaced. Since this is not the case, the lower values for this measure are not surprising.

Last, there is clearly a mismatch between the essential dimensionality of the data and the dimensionality of the SOM. The unidimensionality of the data was by design and it is reflected in the analysis using MDS. Specifically, total score was very closely related to Dimension One of all MDS analyses of each SOM. Dimension Two of these analyses could not be consistently interpreted and appeared to reflect performance on different items in each replication. This suggests that no systematic variance determines the placement of model vectors in the dimension orthogonal to intended class. This raises an important concern regarding the application of SOMs in educational measurement, as to this author's knowledge, two-dimensional SOMs are used exclusively (e.g., Stevens, Ikeda, Casillas, Palacio-Cayetano, & Clyman, 1999; Stevens, Johnson, & Soller, 2005;

Williamson, Bejar, & Sax, 2004). Kohonen (2001) and others (e.g., deBodt, Cottrell, & Verleysen, 2002; Villman, Der, Herrman, & Martinetz, 1997) show that "folds" in the SOM representation of the data will occur when the dimensionality of the data and the SOM do not match. In the case that the SOM has a larger number of dimensions than the data, a linear sequence of model vectors will be maintained only by folding the string of adjacent vectors onto itself in an attempt to occupy the vacant space of the map, much like putting a chain into a box. Since this folding could alter both adjacency and proximity relationships between from the original data, this mismatch may be a primary factor in the low values of projection in the present experiment. Moreover, this hypothesis, if true, would recommend against the apparently common practice of using two-dimensional SOMs for data projection in the absence of information about the intrinsic dimensionality of the data. Evaluating the role of data and map dimensionality is therefore of key importance in assessing the usefulness of SOMs to represent the structure of educational data.

To determine the importance of the match between the dimensionality of the two spaces in the representation of test structure, a second experiment was conducted, and is reported in the next chapter. In this experiment, the dimensionality of the map is maintained, but the dimensionality of the simulated data is increased to two. If the SOM is able to better represent these more complex data, this will be evidence that the match of the dimensionality between the two spaces is an important factor in representing the structure of educational data.

References

de Bodt E., Cottrell M., & Verleysen M. (2002). Statistical tools to assess the reliability of self-organizing maps, *Neural Networks, 15*, 967-978.

Kohonen, T. (2001). *Self-organizing maps* (3rd ed.). Berlin: Springer-Verlag.

Stevens, R., Ikeda, J., Casillas, A., Palacio-Cayetano, J., & Clyman, S. (1999). Artificial neural network-based performance assessments. *Computers in Human Behavior, 15*, 295-313.

Stevens, R., Johnson, D., & Soller, A. (2005). *Cell Biology Education, 4*, 42–57.

Takane, Y., Young, F. W., & de Leeuw, J. (1977). Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika, 42*, 7-67.

Villmann, T. Der, R. Herrmann, M. & Martinetz, T. M. (1997). Topology Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement. *IEEE Transactions on Neural Networks, 8*, 256-266.

Williamson, D. M., Bejar, I. I., & Sax, A. (2004). Automated tools for subject matter expert evaluation of automated scoring. *Applied Measurement in Education, 17*, 323-357.

Chapter 6 - Experiment 2: SOM Representation of Classes Ordered in Two Dimensions

In Experiment One, it was determined that a two-dimensional SOM was limited in its capacity to represent one-dimensional simulated educational test data. Primarily, this limitation was seen in the less than faithful projection of the test data onto the SOM. There were limits in how well the structure in the original data was maintained in the SOM evidenced by, a) the incomplete preservation of adjacency and distance relationships from the metric space to the SOM and, b) the inability to interpret the second dimension both in visual representations of the SOMs and in the analysis of the model vectors using multi-dimensional scaling. It was concluded that these shortcomings might be in part attributable to the difference in dimensionality between the data and the SOM.

To explore the legitimacy of this conclusion, the dimensionality of the simulated test data in the present experiment is increased to two. This change addresses the main concern following from the first experiment, that the mismatch of their respective dimensionality restricted the ability of the SOM to faithfully represent the test data. Furthermore, this change represents a change in the *complexity* of the data; the SOM will have to incorporate data with more complex structure in its representation. As a result, Experiment Two can be seen as a test of the hypothesis that the mismatch in dimensionality decreases SOM representation ability; if performance improves from Experiment One to Experiment Two, it will do so despite increased complexity of the data.

Method

Experiment Two follows the same general form as Experiment One. As mentioned above, the dimensionality of these data has been increased from one to two. This was accomplished by adding 12 items to the original 12 from the first experiment, with the additional items loading on the second dimension. The total number of items for the present experiment is thus 24, with each item loading on one and only one dimension. The responses of the simulated examinees to the additional 12 items represent four ordered classes in Dimension 2, precisely the same way as the responses to 12 items from Experiment One represented four ordered classes of Dimension 1.

The class membership of the simulated examinees for each dimension was determined in the same manner as Experiment One. An equal number of examinees were assigned to each of four classes by setting their ability levels equal to the difficulty of the items representing the ordered class. However, since the class represented by Dimension Two was unrelated to that from Dimension One, there were 16 equiprobable classes representing a complete crossing of four classes from each dimension. All other parameters for Experiment Two are identical to that of Experiment One.

Results

The issues addressed in the present chapter are the same as those in the previous one now in the context of two dimensional data, a) the ability of the SOMs to reveal essential characteristics of the test data and, b) the boundary conditions under which these characteristics are revealed. These issues are examined from the same three perspectives that were used in the previous chapter: a) statistical, b) qualitative and, c) interpretive.

The analyses conducted in each of these three sections also parallel those in Experiment One. The results from those analyses are presented below.

Section I – Statistical Analysis

Map-Data Fit: Quantization Error

A 2 x 3 (SOM Size x Item Discrimination) analysis of variance (ANOVA) was conducted to determine how well the SOM represents the simulated test data in terms of quantization error. The interaction in this analysis was statistically significant, $F(2, 594) = 251.7$, $p < 0.001$, indicating that levels of each main effect were not sufficient to statistically predict QE. As in Experiment One, the practical significance of this result does not follow from its statistical significance given the overwhelming statistical power. An examination of the cell means and standard errors in Table 6.1 underscores this assertion as little deviation from the trend in the main effects are observed in the simple main effects. Both main effects are were also statistically significant, for SOM size, $F(1, 594) = 751015.1$, $p<0.001$, and for Item Discrimination, $F(2, 594) = 1406731.2$, $p<0.001$. The interpretation of these results is consistent with Experiment One; QE decreases with increases in SOM size and with increases in Item Discrimination.

Table 6.1. Mean (Standard Error) Quantization Error by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
4 x 4	1.96 (0.000)	1.68 (0.000)	1.34 (0.001)	1.66 (0.015)
8 x 8	1.70 (0.001)	1.42 (0.000)	1.06 (0.000)	1.39 (0.015)
All	1.83 (0.009)	1.55 (0.009)	1.20 (0.010)	1.53 (0.012)

Projection .

The effect of Item Discrimination and Map Size was examined for the two measures of projection, topological preservation and correlation of distances. Means and standard errors for each measure are presented in Tables 6.2 and 6.3. As has been consistently seen, interactions were statistically significant, $F(2, 594) = 1358.9$, $p<0.001$, and $F(2, 594) = 21.5$, $p<0.001$ for TP and R_{corr}, respectively. The interpretation of this effect for the two measures is quite different, with little improvement in TP being seen with increases in Item Discrimination in the large maps. The same comparison in the small maps reveals large increases in TP, approaching the maximum (97.7%) with Item Discrimination equal to 2.0. In contrast, the interaction effect for R_{corr} can be attributed to a ceiling effect encountered for both large and small maps at the highest level of Item Discrimination, where R_{corr} approached 1.0 (0.96 and 0.94 for small and large maps, respectively).

Table 6.2. Mean (Standard Error) Topological Preservation by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
4 x 4	0.64 (0.003)	0.83 (0.003)	0.98 (0.002)	0.82 (0.008)
8 x 8	0.25 (0.002)	0.27 (0.003)	0.28 (0.003)	0.27 (0.002)
All	0.44 (0.014)	0.55 (0.020)	0.63 (0.025)	0.54 (0.012)

Table 6.3. Mean (Standard Error) Correlations (R_{dist}) between Distances in Metric Space and Associated SOM Co-ordinates by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
4 x 4	0.75 (0.003)	0.90 (0.004)	0.96 (0.004)	0.90 (0.006)
8 x 8	0.58 (0.002)	0.82 (0.005)	0.94 (0.000)	0.83 (0.009)
All	0.68 (0.006)	0.87 (0.004)	0.95 (0.002)	0.87 (0.005)

Both main effects for TP were also statistically significant, $F[1, 594] = 2106.6$, $p<0.001$, and $F[1, 594] = 53679.2$, $p<0.001$ for Item Discrimination and Map Size, respectively. The same main effects were statistically significant for R_{corr}, $F[1, 594] = 4545.7$, $p<0.001$ (Item Discrimination), and $F[1, 594] = 991.0$, $p<0.001$ (Map Size). As in Experiment One, projection was improved when Item Discrimination was higher. Also like the first experiment, there was a difference in TP favouring small maps over large although this difference was particularly large in the present experiment. This same difference was smaller for R_{corr} but reflected a similar pattern. The most notable difference between the two experiments was the size of the values for both measures of projection. In the best condition, (small map, high item discrimination) values of TP and R_{corr} (0.98 and 0.96, respectively) approached their maximums (1.00), indicating that proximity and distance relationships derived from the original data and model vectors, respectively, to the SOM were well preserved. High values for TP were also observed for the small map with Item Discrimination = 1.0 ($\overline{X}_{4x4,\ 1.0} = 0.83$), and for R_{corr} for both map sizes in the two highest Item Discrimination conditions.

Discussion

The pattern of results in Experiment Two followed closely that of Experiment One, again reflecting the impact of map size and item discrimination on the representation of data structure. In general, more discriminating items produced more accurate maps both in terms of model-data fit and projection. The beneficial effect of larger maps was only observed for map-data fit, as smaller maps outperformed larger ones on both measures of projection. The explanation for this is similar to the explanation presented in Experiment One; more units provide greater resolution of the data and thus greater map-data fit. However, the greater number of units in the large maps also decreases the overall likelihood that any two units will be adjacent even when adjacent in the metric space. Small numbers of data points in each receptive field in the larger map also means that the locations of the centroids may be affected by spurious variance in the data. To the extent that this variance in uncorrelated to the two dominant dimensions in the data represented by the co-ordinate dimensions of the SOM, its presence could result in smaller correlations of distance between the two spaces.

Quantization Error across all conditions of Experiment Two was larger than the corresponding values in Experiment One. This may reflect the increased complexity of the two-dimensional versus one-dimensional data. In effect, it may be a sign that the original data in Experiment Two populates a greater dimensional space than those in Experiment One, and consequently that each cluster must represent a larger region. Large values on the measures of projection in Experiment Two provide evidence for the hypothesis generated from Experiment One; SOMs better represent the structure of data when there is a match between the dimensionality of those data and of the map. This is

evidenced particularly for the small map, high item discrimination condition but was also evident in the comparisons of cell and marginal means for TP and R_{dist} between the two experiments; improvements in projection were seen in Experiment Two for all but one comparison. This is further evidence for the central role of dimensional match in creating interpretable SOMs.

Section II – Qualitative Examination of the SOM

Visual Examination of SOM Solutions

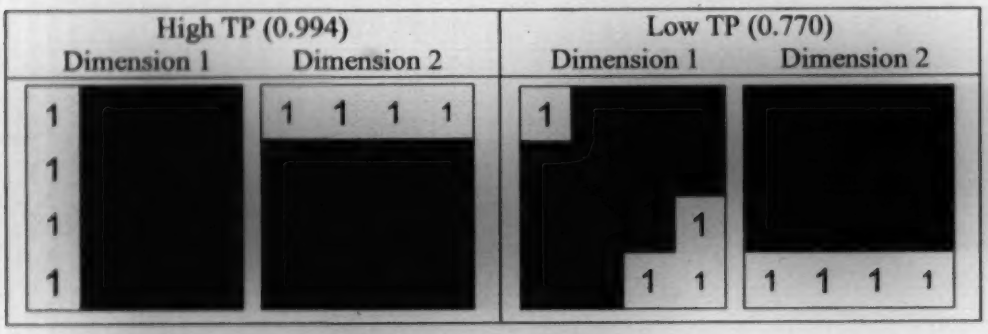
The interpretation of the statistical results presented in Section I is enhanced by the visual examination of SOMs from each condition. Figure 6.1 displays the modal class for each unit and for each dimension for two replications of the small map, high item discrimination condition with the highest and the lowest values for TP[5]. In order to clearly show how the two data dimensions were represented in the SOM, each map shows the modal class for data dimensions one and two in the left- and right-hand, respectively for each replication. The size of the numbers representing each model class is proportional to the number of observations classified to that unit.

One striking feature of the SOM having high TP is the clarity with which it renders the structure of the data. The simulated classes in each dimension are grouped together and the classes are ordered from lowest to highest across the surface of the map. Furthermore, the orthogonality of the dimensions is preserved in the map; Dimension 1 is

[5] There are two ways to calculate modal intended class for each unit for the purposes of these figures. The first is to determine which of the sixteen simulated classes was most frequently classified to the unit. Each of these 16 classes is uniquely identified by one of four classes in each dimension, and these are the values included in the figure. For example, the modal class for unit 7 (3,2) on the High TP was class 10. This class was simulated to be of Class 3 on Dimension 1 and Class 2 on Dimension 2 and thus for unit 7, the numbers 3 and 2 appear in the left- and right-hand figures of the High TP replication in the above figure. The second way is to determine the modal class separately for each dimension, ignoring the class of the other dimension. In this case, the decision for the modal class of each dimension is between four, not sixteen classes, but two separate decisions are made. Though the two methods lead to similar results, the first method was used exclusively.

ordered from left to right whereas Dimension 2 is order from top to bottom. This figure indicates that under these conditions, the SOM can create representations of simulated test data that are clearly interpretable. Interestingly, though derived from the same data, the two replications presented here differ considerably in their interpretability. Though Dimension 2 from the replication with low TP is well ordered in the map, it fails to clearly display the simulated structure of the Dimension 1. This demonstrates that the SOM cannot be relied upon to consistently produce faithful reproductions of the test data. However, it is worthy of note that this replication was highly atypical of this condition. Only two replications had topological preservation less than 0.95, the low TP replication above (0.77), and one other (TP = 0.89). It therefore may be argued that in the large majority of replications, good solutions will be obtained.

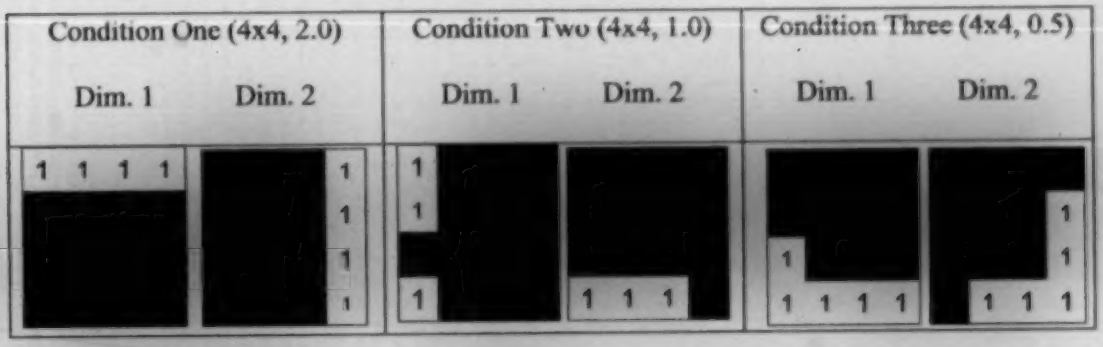
Figure 6.1. Modal Simulated Class Membership for Two Self-Organizing Maps in Experiment Two, Small Map, Item Discrimination = 2.0.



Examining typical replications from each condition further illuminates the capacity of the two-dimensional SOM to yield interpretable representations of two-dimensional data. Figure 6.2 displays SOMs from one typical replication that was randomly chosen from each condition, 1 through 3. The definition of typical replication remains the same from

that as Experiment One; a replication is considered typical if its associated values for QE, TP, and R_{corr} are within one standard unit from the mean of the condition.

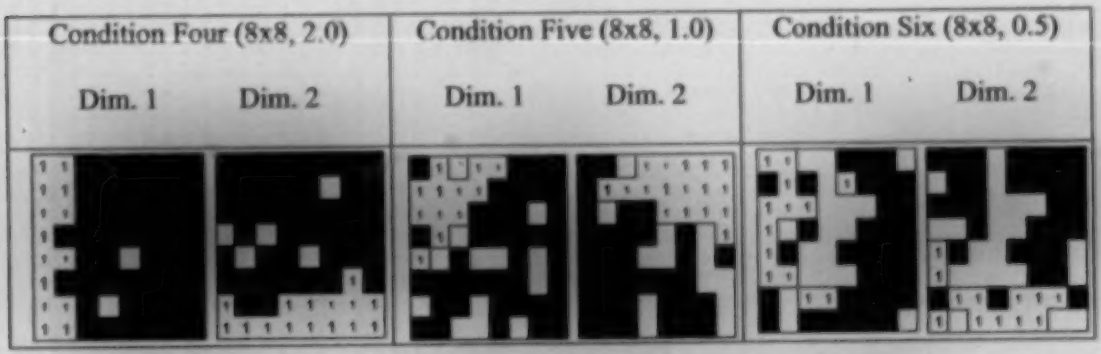
Figure 6.2. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Two, Conditions One, Two, and Three.



The SOM representations of the typical replications from the three conditions differ markedly with changes in Item Discrimination. In Condition One, the structure of the input data is very clearly depicted. Classes within each dimension are represented by contiguous units, adjacent units belong to either the same class or to a class representing the next most or next least able examinees, the classes are correctly ordered, and the two orthogonal dimensions in the data are represented by orthogonal dimensions in the map. Condition Two reveals many of these properties, but less consistently. Assuming that each column in Dimension One and each row in Dimension Two represents a single class within its respective dimension, there are a total of seven deviations from the ideal pattern exhibited in Condition One. The map from the typical replication of Condition Three reveals little of the simulated structure of the data. Dimension 1 appears to be somewhat preserved, as Classes One and Four are separate and occupy the extreme top and bottom of the map. Beyond the contiguity of Class One, the representation of Dimension 2 is

very limited. No ordering is evident, classes of neighbouring ability are not consistently adjacent in the map, and single classes do not occupy contiguous regions.

Figure 6.3. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Two, Conditions Four, Five, and Six.



Note: Map locations are blank for which more than one modal class was possible.

As in the small map conditions, large maps vary in interpretability with item discrimination. Condition Four (Item Discrimination = 2.0) has the most clearly interpretable structure, very similar to Condition One. Each dimension is represented by a co-ordinate axis in the map along which the classes are well ordered. Condition Five (Item Discrimination = 1.0) has some discernable ordering as in general, classes representing the extreme ability groups were located far away from each other. Interestingly, the ordering within the dimensions seems to be most prevalent along the diagonal as opposed to the co-ordinate axes. Also, approximately 20% of the map locations have no clear modal class, possibly due to the small number of observations in each receptive field. Condition Six (Item Discrimination = 0.5) has many of these class-indeterminate map locations and less overall discernable structure. The left- and right-

hand regions of Dimension One seem to correspond to Classes One and Four, respectively although these classes are not restricted to these regions. Top and bottom regions in Dimension Two also appear to represent extreme classes while Classes Two and Three are interspersed throughout the SOM.

Discussion

There were clear improvements in the interpretability of maps in Experiment Two versus Experiment One. First, co-ordinate axes from SOMs in Experiment One did not exclusively represent one dimension in the data. However, this *was* clearly observed in Experiment Two, particularly in small map and high item discrimination conditions. Second, the linear ordering of classes was represented linearly in maps from Experiment Two. In Experiment One, even for the most favourable conditions (small map, high item discrimination), class order was often folded in the map. Both of these points support the critical role of dimensional match in generating interpretable maps.

Section III – Interpretation of the SOM

In Experiment One, Multi-Dimensional Scaling (MDS) was used to identify the characteristics of test- and item-level performance that were represented by the SOM. This analysis was of particular interest since measures of projection and visual examinations of the SOMs showed that the co-ordinate axes of the map did not and could not clearly represent the dimensional characteristics of the data. In the present experiment, the co-ordinate axes appear to have a clear interpretation; each axis seems to represent performance on each of the two ability dimensions in the data. MDS for the present experiment will therefore be used to provide further evidence for this

interpretation of the SOM solution.

Table 6.4. Stress1 and Variance Accounted For (VAF) in One- and Two-Dimensional MDS Analyses on Typical Replications in Each Condition

	MDS Analysis			
	One-Dimensional		Two-Dimensional	
Condition	Stress1	VAF	Stress1	VAF
1 (4x4, $a = 2.0$)	0.376	0.543	0.040	0.989
2 (4x4, $a = 1.0$)	0.422	0.441	0.067	0.969
3 (4x4, $a = 0.5$)	0.407	0.460	0.156	0.829
4 (8x8, $a = 2.0$)	0.407	0.509	0.079	0.964
5 (8x8, $a = 1.0$)	0.409	0.501	0.147	0.877
6 (8x8, $a = 0.5$)	0.506	0.280	0.281	0.552

As in Experiment One, each of the typical replications from Section II was analyzed using ALSCAL. The values of stress and the variance accounted for (VAF) in each replication for one and two dimensions are listed in Table 6.4. The interpretation of these data are much the same as in Experiment One; higher item discrimination leads to better fit, larger maps are more difficult to fit than smaller ones, and two-dimensional solutions are better-fitting than one-dimensional solutions.

The question of what the dimensions identified by MDS reflect about the SOM solution is addressed in Table 6.5. Here, the correlation of average score for observations at each unit with the MDS co-ordinates is presented. For the one-dimensional analysis,

the correlations presented are between the average *total* score and the MDS co-ordinates at each unit. The correlations presented for the two-dimensional solution are between the *subtest* scores for each group of 12 dimensionally coherent items and the MDS co-ordinates for each unit for each dimension. In order to maximize the interpretability of these co-ordinates in terms of the dimensions in the data, MDS co-ordinates were rotated to yield the maximum absolute correlation with the subtest scores. These correlations are the values presented in Table 6.5.

Table 6.5. Correlation between Dimensional Co-ordinates of Each SOM Unit in MDS Analyses and Total Expected Score for Unit Members, Performed Separately for Typical Replications of Each Condition

	MDS Analysis		
	One-Dimensional	Two-Dimensional	
Condition	Dim 1	Dim 1	Dim 2
1 (4x4, $a = 2.0$)	0.719	0.997	-0.997
2 (4x4, $a = 1.0$)	0.950	-0.899	0.879
3 (4x4, $a = 0.5$)	0.170	0.967	-0.967
4 (8x8, $a = 2.0$)	0.738	0.875	0.874
5 (8x8, $a = 1.0$)	0.764	-0.991	-0.987
6 (8x8, $a = 0.5$)	0.199	-0.760	0.724

From Table 6.5 it can be seen that the two-dimensional MDS co-ordinates are very closely related to the 12-item subtest scores for each unit. The large values of these

correlations indicate that the placement of the model vectors is almost entirely determined by the two-dimensional structure of the data represented by the subtest scores. Therefore, the MDS analysis reflects the clarity of the SOM's representation of the test data, despite changes in map size and item discrimination.

It is notable that correlations of the one-dimensional co-ordinates with the total test score were also large. This can be explained by noting that only a subset of units require two dimensions to fully describe their performance; for example, extremely high average scores can only result from high values on each of the two simulated classes. By way of analogy, there is only one combination of scores from two dice that yields a score of twelve; the sum of the die faces uniquely identifies the value shown by each die. However, a score of seven does not reveal the results at each die. Similarly, a total test score between the two extremes does not contain information to determine the combination of classes that produced it. Therefore, though the correlations are high for one dimension, this single dimension is insufficient to represent the complexity of the data. Two dimensions are needed to clearly depict the structure of the data.

General Discussion

This experiment was designed to test the hypothesis that a match between the dimensionality of the SOM and the test data being analyzed could maximize the conformity of the SOM representation with the characteristics of the original data. This experiment can be interpreted as evidence for the correctness of that hypothesis. Despite an increase in the complexity of the data in the present experiment when compared with those of Experiment One, the essential characteristics of the data appear to be more clearly rendered. This conclusion follows from evidence from each of the three

perspectives presented, statistical, qualitative, and interpretive. The statistical evidence consists of larger absolute values on measures of projection, particularly for the small map condition. This seems to indicate that the two-dimensional SOM is preserving relationships of the original data better when they themselves are two-dimensional. Qualitative evidence follows from the visual inspection of specific SOMs; regions of the maps appeared to be more tuned to individual classes than in Experiment One. Furthermore, since the map was the same dimensionality as the data, it was a natural medium to represent the data; each of the co-ordinate axes in the map came to represent performance on each dimension. Last, MDS analyses revealed the inherent dimensionality of the set of model vectors. The two-dimensional nature of the representation of the test data was strongly present in this analysis, indicating that the SOM was selectively sensitive to it.

Though all these sources of evidence are consistent with the hypothesis, there are several caveats in this interpretation that bear mention and that can be addressed empirically. First, in the research presented thus far, only the two-dimensional SOM has been considered. If the hypothesis is a general one, it ought to apply to SOMs of dimensionality other than two. Second, no direct statistical evidence was provided for the superiority of maps with the same dimensionality as the test data versus those with different dimensionalities. Furthermore, in the case of mismatches, only the case where the dimensionality of map exceeds that of the data has thus far been considered. The opposite case, where the dimensionality of the data *exceeds* that of the SOM has not been examined.

In order to address these potential shortcomings, a further series of simulations and subsequent analyses were undertaken, and are reported in the next chapter. These simulations examine more completely the issue of dimensional match between SOM and test data by employing one-dimensional SOMs to represent both one- and two-dimensional data. Adding these further simulations, an additional factor can be considered in statistically analyzing the results of Experiments One and Two: dimensional match. This statistical analysis should provide more compelling evidence for the critical role of this factor in using SOMs to represent test data.

Chapter Seven – Experiments 3 and 4: One-Dimensional SOMs

The results from Experiments One and Two indicate the importance of a match of dimensionality between data and SOM in order to create 'better' maps. The appearance of higher values of projection and more interpretable maps in Experiment Two versus One demonstrates this. This has direct implications for using SOMs as a method to determine test structure; dimensional structure of a test will be most clearly represented with a map of the same dimensionality. Experiment Three is designed as a direct test of this hypothesis.

In order to more definitively determine the role of the dimensional match, a statistical analysis is conducted comparing the dependent measures under conditions of match versus non-match. Before this analysis is conducted, however, the results from Experiments One and Two will be extended by the inclusion of one-dimensional maps. Since the data in the previous two experiments were one- and two-dimensional, a complete analysis of these data from the perspective of dimensional match must include both one- and two-dimensional maps. The goals of Experiment Three are therefore twofold. First, it is designed to better understand the capabilities of a one-dimensional SOM in representing simulated educational test data in light of the potentially important role for dimensional match. This is an important goal in itself as it is often an explicit goal of test developers to build unidimensional tests and if dimensional match is important then one-dimensional SOMs ought to represent those tests most effectively. The second goal is to test *statistically* the impact of dimensional match for the representation of the structure of educational test data. This will be accomplished by

coding all one-and two-dimensional data and maps in terms of dimensional match and using this coding to predict the three measure of SOM quality: QE, TP, and R_{dist}.

The first part of Experiment Three follows the same form as the two previous experiments; examined are a) the ability of the SOM to reveal essential characteristics of the test data and, b) the boundary conditions under which those characteristics are revealed. These two issues are again examined from the statistical, qualitative, and interpretive perspectives, in separate sections. These analyses are first done with one-dimensional maps and one-dimensional data, and then repeated with one-dimensional maps and two-dimensional data. The final section of this chapter features a statistical analysis of the specific role for dimensional match using data from all previous experiments. At the conclusion of this chapter a role for dimensional match is discussed.

Experiment 3: Data and Method

The same data used in Experiment One were used in Experiment Three. The SOM comprised 16 units, the same number as in Experiment One, but the units were organized as a 16 by 1 array rather than a 4 by 4 array for the small map condition, and as a 64 by 1 rather than an 8 x 8 array for the large map condition. Otherwise, training proceeded in the same manner as Experiment One.

Results

Section I – Statistical Analysis

Quantization Error

A 2 x 3 (SOM Size x Item Discrimination) analysis of variance (ANOVA) was conducted on QE. The interaction between these factors was statistically significant, $F(2, 594) = 1314.2$, $p < 0.001$, indicating that differences in QE resulting from map size varied

with item discrimination. In general, greater differences in QE were observed between large and small maps when item discrimination was greater. Main effects were also significant for SOM size, $F(1, 594) = 187554.6$, $p<0.001$, and for Item Discrimination, $F(2, 594) = 238035.8$, $p<0.001$ revealing a familiar pattern of larger QE for smaller maps and higher item discriminations. Means for all conditions are presented in Table 7.1.

Table 7.1. Mean (Standard Error) Quantization Error by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
1 x 16	1.21 (0.000)	0.94 (0.000)	0.61 (0.001)	0.92 (0.014)
1 x 64	0.92 (0.001)	0.63 (0.001)	0.23 (0.001)	0.59 (0.016)
All	1.06 (0.010)	0.78 (0.011)	0.42 (0.014)	0.76 (0.013)

Projection

An ANOVA with the same factors was also conducted for both TP and R_{dist} (means are presented in Tables 7.2 and 7.3). Both analyses revealed statistically significant interaction effects, $F(2, 594) = 72.2$, $p<0.001$, and $F(2, 594) = 25.4$, $p<0.001$ for TP and R_{dist}, respectively. Both interactions are interpreted similarly; differences in projection between small and large maps decrease as item discrimination increases. Main effects for each measure were also statistically significant. The main effect of map size shows that projection is better for smaller maps than larger ones ($F(1, 594) = 347.5$, $p<0.001$, and $F(1, 594) = 1514.3$, $p<0.001$ for TP and R_{dist}, respectively) while the main effect of item discrimination shows that high item discriminations have positive effects on measures of

projection ($F(2, 594) = 667.0$, $p<0.001$, and $F(2, 594) = 6128.9$, $p<0.001$ for TP and R_{dist}, respectively). This pattern of results is the same as observed in the previous two experiments.

Table 7.2. Mean (Standard Error) Topological Preservation by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
1 x 16	0.50 (0.005)	0.58 (0.006)	0.64 (0.007)	0.57 (0.005)
1 x 64	0.37 (0.003)	0.46 (0.005)	0.63 (0.007)	0.49 (0.013)
All	0.44 (0.005)	0.52 (0.006)	0.64 (0.005)	0.53 (0.005)

Table 7.3. Mean (Standard Error) Correlations (R_{dist}) between Distances in Metric Space and Associated SOM Co-ordinates by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
1 x 16	0.58 (0.006)	0.83 (0.004)	0.91 (0.001)	0.81 (0.009)
1 x 64	0.44 (0.006)	0.71 (0.005)	0.86 (0.001)	0.71 (0.010)
All	0.51 (0.006)	0.77 (0.005)	0.89 (0.002)	0.76 (0.007)

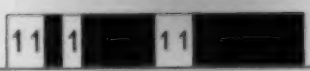
The results above show that map size and item discrimination have a similar effect with one-dimensional as with two-dimensional maps. In particular, they show that larger sized maps lead to lower values of QE whereas for measures of projection, smaller maps

are superior. High item discriminations lead to higher quality maps, regardless of the measure examined.

Section II – Qualitative Examination of SOMs

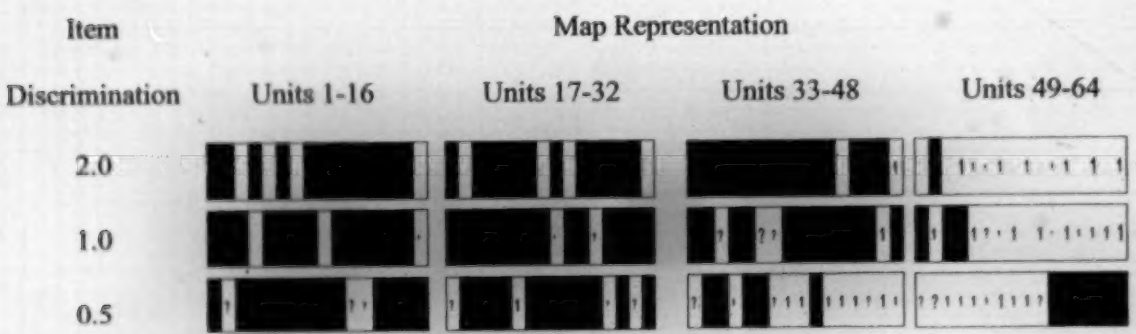
Visual representations of the maps shed further light on the nature of the representation of one-dimensional data with a one-dimensional SOM. Figure 7.1 shows such representations from typical replications of each of the small map conditions. This figure corroborates the results from the statistical analysis of projection; when item discriminations are high, better maps are produced. This can be clearly seen from the superiority of the first map in rendering the test structure as compared with the second and third maps. The first map represents classes in precise order from lowest- to highest-achieving, from left to right. The second map has clear order, with one exception; the third unit from the left represents Class 2, while its neighbours both represent Class 1. The third map, representing a typical replication from the condition of the lowest item discrimination, has many such misplaced units. Importantly, and unlike the maps produced in Experiment One, visual representations of maps with high item discriminations are clearly represented as unidimensionally ordered.

Figure 7.1. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Three, Small Maps Only.

Item Discrimination	Map Representation
2.0	
1.0	
0.5	

In contrast, even with high item discriminations, large maps do not as clearly represent the test structure for these data as do small maps (see Figure 7.2). Of the three conditions, the map trained with the data of the highest item discrimination shows most clearly the intended unidimensionally ordered structure but many units are either vacant (i.e., no observations were found in their receptive fields) or were misplaced (i.e., were deviations from the strict ordering). As item discrimination decreases, less vacant units are observed, but more misplaced units appear, as well as those whose intended class is indeterminate (denoted with a '?'). In addition, for the typical replication with the lowest item discrimination, overall ordering of the data in the map appears to break down in the extreme right of the SOM. That the structure of the data is not well represented by these large maps may reflect over-fitting of the data; too few data points are represented by units in the maps.

Figure 7.2. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Three, Large Maps Only.



Note: '?' denotes units with more than one most frequent intended class.

Section III – Interpretation of the SOM

From visual inspection of typical maps it can be seen that in some conditions, the SOM clearly depicts the dominant dimension in the data. This can be seen by observing that the ordering of model vectors is consistent with the order of the simulated classes. What remains to be determined is how well the SOM has extracted the dominant dimension from the data and also the extent to which the ordering reflects properties of the data directly related to class membership.

Table 7.4. Stress1 and Proportion of Variance Accounted For (VAF) in One- and Two-Dimensional MDS Analyses on Typical Replications in Each Condition

	MDS Analysis			
	One-Dimensional		Two-Dimensional	
Condition	Stress1	VAF	Stress1	VAF
1 (16, a = 2.0)	0.092	0.972	0.074	0.979
2 (16, a = 1.0)	0.153	0.923	0.108	0.948
3 (16, a = 0.5)	0.336	0.654	0.227	0.716
4 (64, a = 2.0)	0.175	0.908	0.119	0.945
5 (64, a = 1.0)	0.293	0.745	0.205	0.805
6 (64, a = 0.5)	0.439	0.428	0.294	0.521

To address these questions, each of the typical replications from above was analyzed using MDS. As in the previous experiments, the primary interest is the projection of the model vectors in two dimensions and therefore MDS analyses are

reported for one and two dimensions only. The values of stress and the proportion of variance accounted for (VAF) in each replication for one and two dimensions are listed in Table 7.4. Two trends emerge from these results. First, as item discrimination decreases, values of stress increase and the variance accounted for decreases. Second, larger maps are more difficult to fit, as evidenced by the larger values of stress and lower amount of variance accounted for as compared with the smaller maps. It is interesting to note that the results from the MDS correspond well to the quality of visual representation from Section II; the most clearly interpretable visual representations (i.e., small map conditions with item discrimination = 2.0, 1.0, and large map condition with item discrimination = 2.0) had the lowest values of stress and the highest values of VAF.

To interpret the dimensions identified by MDS in terms of test and item performance, a correlation analysis was performed between the co-ordinates determined by MDS and the average total score on the test for all examinees in the same receptive field (see Table 7.5). Recall that this average was computed by summing together the elements of the model vectors. Correlations approaching 1.0 were observed in every condition. This analysis shows clearly that the most dominant dimension represented by the model vectors and identified by MDS corresponds unequivocally to a dominant dimension in the data.

From the above results, it appears that one-dimensional maps represent well one-dimensional data, particularly in the small map and high item discrimination conditions. This is evidenced by high values of projection, interpretable visual representation, and clear representation of the dominant structure in the data for maps in these conditions. However, this conclusion remains tentative before direct comparison with other

conditions in which dimensional match does not hold. The following section examines such data, those resulting from the analysis of two-dimensional data with the one-dimensional SOM.

Table 7.5. Correlation between Dimensional Co-ordinates of Each SOM Unit in MDS Analyses and Total Expected Score for Unit Members, Performed Separately for Typical Replications of Each Condition

	MDS Analysis		
	One-Dimensional	Two-Dimensional	
Condition	Dim 1	Dim 1	Dim 2
1 (16, a = 2.0)	-0.999	-0.998	-0.255
2 (16, a = 1.0)	0.999	0.998	0.004
3 (16, a = 0.5)	-0.993	-0.993	-0.006
4 (64, a = 2.0)	0.999	0.996	0.512
5 (64, a = 1.0)	0.996	0.998	0.080
6 (64, a = 0.5)	-0.946	0.974	0.189

Experiment 4: Data and Method

The same two-dimensional data as used in Experiment Two were used in Experiment Four. The SOMs comprised 16 and 64 units for the small and large map conditions, respectively. These were the same as in Experiment Three. Otherwise, training proceeded in the same manner as Experiment Two.

Results

Section I – Statistical Analysis

Quantization Error

In order to determine the impact of analyzing two-dimensional test data with a one-dimensional SOM, a series of 2 x 3 (SOM Size x Item Discrimination) analyses of variance (ANOVA) were conducted. Like the previous experiments, the first of these analyses focused on QE, the measure of map-data fit. For this measure, the interaction between the factors was statistically significant, $F(2, 594) = 121.8$, $p < 0.001$, indicating that differences in QE resulting from map size were slightly larger with increases in item discrimination. Main effects were also significant for both SOM size, $F(1, 594) = 593032.8$, $p<0.001$, and for Item Discrimination, $F(2, 594) = 1143476.0$, $p<0.001$ revealing that QE was larger for smaller maps and lower item discriminations. Means and standard errors are presented in Table 7.6.

Table 7.6. Mean (Standard Error) Quantization Error by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
1 x 16	1.96 (0.000)	1.68 (0.000)	1.34 (0.000)	1.66 (0.015)
1 x 64	1.70 (0.001)	1.42 (0.001)	1.06 (0.000)	1.40 (0.015)
All	1.83 (0.009)	1.55 (0.009)	1.20 (0.010)	1.53 (0.012)

Projection

Means for the ANOVAs for TP and R_{dist} are displayed in Tables 7.7 and 7.8, respectively. The ANOVA for TP revealed no statistically significant interaction between map size and item discrimination ($F(2, 594) = 1.202$, $p=0.301$), while this interaction was significant for R_{dist}, $F(2, 594) = 72.0$, $p<0.001$. For TP, the lack of interaction demonstrated no differential effect of map size over levels of item discrimination, while for R_{dist}, the significant interaction was due to a *narrowing* of difference between small and large maps as item discrimination increased. This seems to reveal some kind of ceiling effect, around the low value of 0.6. Both main effects for both measures were statistically significant. The main effect of map size for TP and R_{dist} ($F(1, 594) = 1623.8$, $p<0.001$, and $F(1, 594) = 1114.0$, $p<0.001$, respectively), shows that small maps represent relationships between data better than large maps. The significant main effects of item discrimination ($F(2, 594) = 763.3$, $p<0.001$ and $F(2, 594) = 773.9$, $p<0.001$ for TP and R_{dist}, respectively) shows projection is improved when item discriminations are high.

Table 7.7. Mean (Standard Error) Topological Preservation by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
1 x 16	0.49 (0.003)	0.53 (0.004)	0.64 (0.004)	0.55 (0.004)
1 x 64	0.37 (0.003)	0.43 (0.003)	0.51 (0.003)	0.44 (0.004)
All	0.43 (0.005)	0.48 (0.004)	0.57 (0.005)	0.49 (0.004)

Table 7.8. Mean (Standard Error) Correlations (R_{dist}) between Distances in Metric Space and Associated SOM Co-ordinates by Condition

	Item Discrimination			
Map Size	0.5	1.0	2.0	All
1 x 16	0.54 (0.003)	0.59 (0.004)	0.61 (0.002)	0.58 (0.002)
1 x 64	0.41 (0.002)	0.52 (0.003)	0.57 (0.002)	0.50 (0.004)
All	0.48 (0.005)	0.56 (0.004)	0.59 (0.002)	0.54 (0.003)

The most notable characteristic of the two analyses of projection is that, relative to other experiments, the measures of projection are uniformly low. This may indicate the negative effect of the mismatching of dimensionality between data and map, a possibility explored further below.

Section II – Qualitative Examination of SOMs

The visual representation of two-dimensional data with a one-dimensional map helps further highlight the impact of the dimensional mismatch. Figure 7.3 shows these representations from typical replications of the small map conditions. The most striking characteristic of this figure is the apparent lack of consistent ordering in the SOM for each of the dimensions. Looking more closely, however, it can be seen that some *local* ordering exists, since in the two conditions of highest item discrimination, adjacent units always represent classes that follow each other in ability. Even in the condition of lowest item discrimination there are only two pairs of adjacent units that represent classes whose difference in number is two. However, this local ordering does little to elucidate the test structure.

Figure 7.3. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Four, Conditions One, Two, and Three.

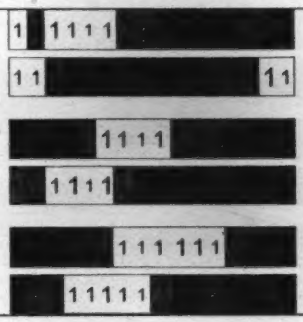
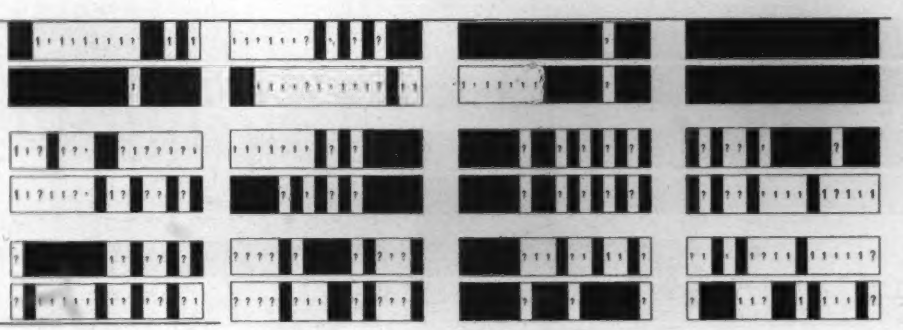
Item Discrimination	Map Representation
2.0	
1.0	
0.5	

Figure 7.4. Most Frequent Intended Class Membership for Typical Self-Organizing Maps in Experiment Three, Conditions Four, Five, and Six.

Item Discrimination	Map Representation			
	Units 1-16	Units 17-32	Units 33-48	Units 49-64
2.0				
1.0				
0.5				

The lack of ordering is more dramatic for large maps. With the exception of when item discrimination is 2.0, it is difficult to determine any regularities and any relationship of map structure to test structure. In the condition with high item discrimination, there are some contiguous regions where class remains consistent, but even here there is little discernable order.

Section III – Interpretation of the SOM

Given the limited ability of the one-dimensional SOM to represent visually the structure of two-dimensional data, an interpretation of the test structure implied by the SOM is of limited value. However, if the SOM is still able to shed light on the dimensionality of the data through the placement of the model vectors, it could be argued that the SOM may still be informative about test structure, even when there is dimensional mismatch. That is, the location of the model vectors themselves and not their association with co-ordinate locations in the SOM may be central in evaluating the dimensionality of the test.

Table 7.9. Stress1 and Proportion of Variance Accounted For (VAF) in One- and Two-Dimensional MDS Analyses on the Model Vectors from Typical Replications in Each Condition in Experiment 4

	MDS Analysis			
	One-Dimensional		Two-Dimensional	
Condition	Stress1	VAF	Stress1	VAF
1 (16, $a = 2.0$)	0.375	0.556	0.036	0.991
2 (16, $a = 1.0$)	0.394	0.509	0.074	0.964
3 (16, $a = 0.5$)	0.439	0.407	0.160	0.827
4 (64, $a = 2.0$)	0.391	0.549	0.082	0.962
5 (64, $a = 1.0$)	0.424	0.468	0.141	0.888
6 (64, $a = 0.5$)	0.475	0.340	0.261	0.613

In order to address this possibility, each of the typical replications from above was analyzed using MDS. The values of stress and the proportion of variance accounted for (VAF) in each replication for one and two dimensions are listed in Table 7.9. The most significant finding from this analysis is the much improved fit of the two-dimensional analysis. This clearly indicates that the model vectors represent well the dimensionality of the data, despite the one-dimension map.

Table 7.10. Correlation between MDS Dimensional Co-ordinates of Each SOM Unit and Total Expected Score by Subscale for Unit Members, Performed Separately for Typical Replications of Each Condition in Experiment 4

	MDS Analysis	
Condition	Dim 1	Dim 2
1 (16, $a = 2.0$)	-0.999	-0.999
2 (16, $a = 1.0$)	0.999	-0.999
3 (16, $a = 0.5$)	-0.991	-0.989
4 (64, $a = 2.0$)	-0.997	-0.997
5 (64, $a = 1.0$)	-0.995	-0.996
6 (64, $a = 0.5$)	-0.973	-0.963

To determine whether the dimensions identified by MDS relate to the characteristics of the items, a correlation analysis was performed between the co-ordinates determined by the two-dimensional MDS analysis and the mean subtest scores for all examinees in the same receptive field (see Table 7.10). MDS co-ordinates were

rotated in order to obtain the maximum correlation. The correlation between subtest scores and the rotated MDS co-ordinates approached 1.0 for each dimension in every condition. This analysis shows clearly that the dominant dimensions represented by the model vectors are the dominant dimensions in the data. This is notable, since the map itself was one- and not two-dimensional. It appears as though the fit of the model vectors to the data depends very little upon their actual arrangement in the map.

Discussion – Experiments 3 and 4

Experiments Three and Four seem to demonstrate that a match of dimensionality between data and SOM has a significant impact on projection. The lower values for TP and R_{dist} and the lack of discernable order in the visual representations of typical replications in Experiment 4 appear to demonstrate this. Less clear is the impact of dimension match on map-data fit. Since QE appears to be dependent on the complexity of the data, a determination of the effect of dimensional match on this aspect of SOM quality must also await a comparison between conditions that have equally complex data, but differ on dimensional match.

In the last section of this chapter, an analysis of data from all four experiments presented thus far is undertaken. In particular, a comparison is made between all 3 statistical measures of SOM quality on the basis of dimension match. Two outstanding issues are intended to be addressed by this analysis. Measures of projection appear improved when there is dimensional match but this apparent improvement has not yet been confirmed statistically. Second, the role of dimensional match is unclear for Quantization Error since no direct comparison has been made when the complex of the data and the size of the map are held constant. These two factors appear to be the most

salient determinants of QE. By combining data across all conditions, these two issues can be directly addressed.

Combined Analysis of All Experiments

Quantization Error

A 3 x 2 x 2 (Item Discrimination x Map Size x Dimensional Match) ANOVA was conducted to determine the role of each factor in the prediction of QE. Table 7.11 shows the cell means for each combination of factors, and Figures 8.5 and 8.6 display these means for small and large map conditions, respectively. No interactions were statistically significant, indicating that the differences in QE between small maps and large maps did not depend on levels of item discrimination. Examining the main effects, Item Discrimination and Map Size were statistically significant, $F(2, 2388) = 537.0$, $p<0.001$ and $F(1, 2388) = 354.7$, $p<0.001$, respectively. The main effect of dimensional match was not statistically significant. This is corroborated by Table 7.11 and Figures 7.5 and 7.6; no differences by dimensional match are apparent.

Table 7.11. Mean Quantization Error across All Conditions by Map Size, Item Discrimination, and Dimensional Match

		Item Discrimination		
Map Size		0.5	1.0	2.0
4 x 4	No Match	1.58	1.31	0.98
	Match	1.58	1.31	0.98
8 x 8	No Match	1.31	1.03	0.66
	Match	1.31	1.02	0.65

Figure 7.5. Mean Quantization Error by Item Discrimination and Dimensional Match for All Small Map Conditions.

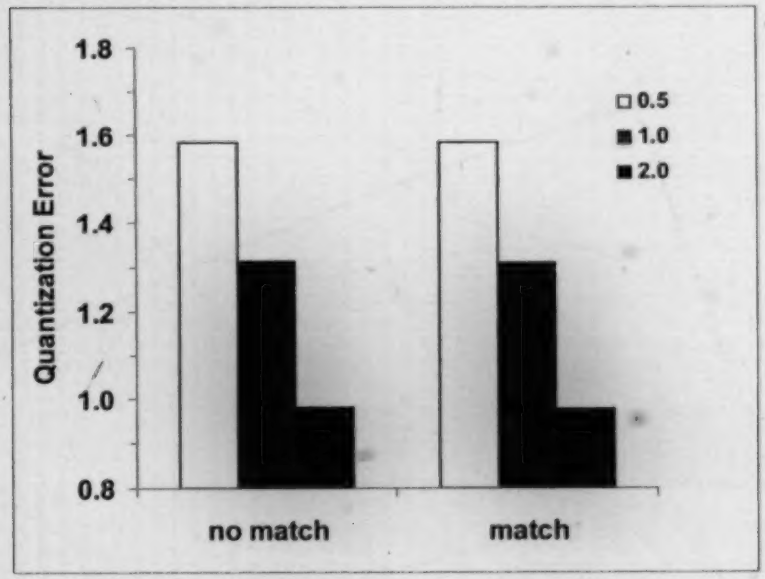
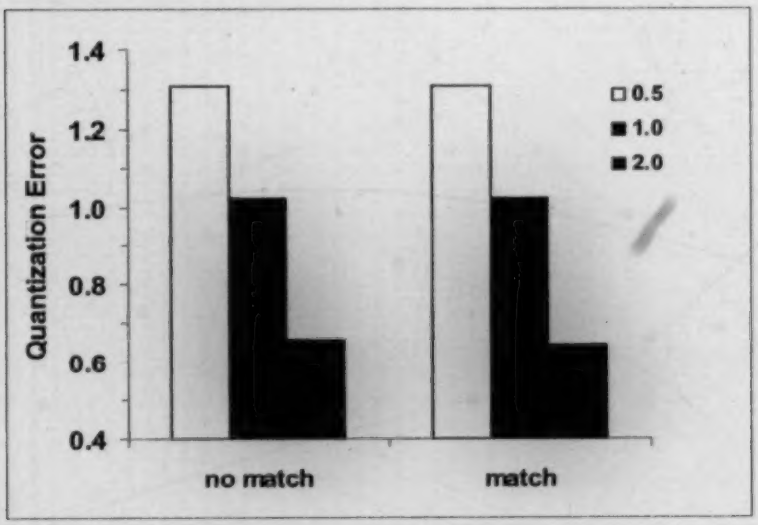


Figure 7.6. Mean Quantization Error by Item Discrimination and Dimensional Match for All Large Map Conditions.



Topological Preservation

A 3 x 2 x 2 (Item Discrimination x Map Size x Dimensional Match) ANOVA was conducted to determine the role of each factor in the prediction of TP. Table 7.12 shows the cell means for each combination of factors, and Figures 7.7 and 7.8 display these means for small and large map conditions, respectively. Each of the interaction effects and main effects were statistically significant. The 3-way interaction of all factors ($F(2, 2388) = 3.4, p<0.05$), indicates that the difference between match and non-match conditions depends upon both map size and item discrimination. Focusing on the interaction of Match and Item Discrimination ($F(2, 2388) = 11.8, p<0.001$), it can be seen that TP is greater for match than for non-match conditions for item discrimination 1.0 or greater. For lower item discrimination, match appears to have little effect. The interaction between Map Size and Match ($F(1, 2388) = 8.7, p<0.005$) reveals that increases in TP due to match are more modest for large than for small maps.

Table 7.12. Mean Topological Preservation Across All Conditions by Map Size, Item Discrimination, and Dimensional Match

		Item Discrimination		
Map Size		0.5	1.0	2.0
4 x 4	No Match	0.57	0.64	0.73
	Match	0.57	0.70	0.81
8 x 8	No Match	0.30	0.35	0.43
	Match	0.31	0.37	0.46

Figure 7.7. Mean Topological Preservation by Item Discrimination and Dimensional Match for All Small Map Conditions.

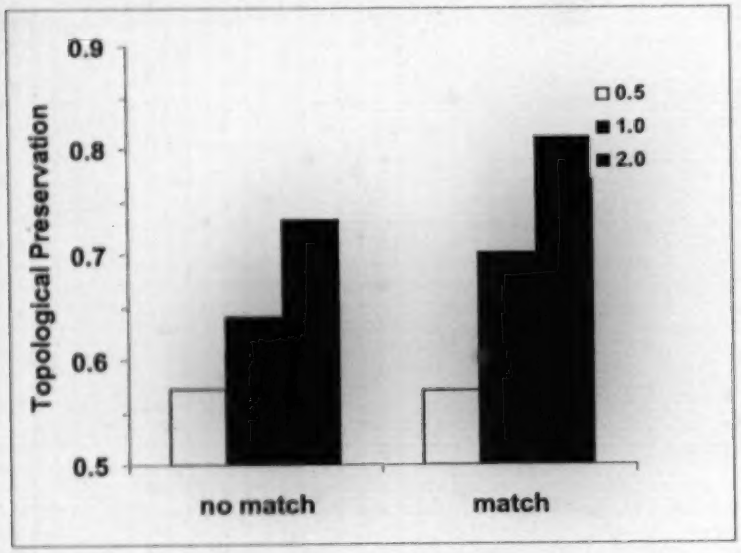
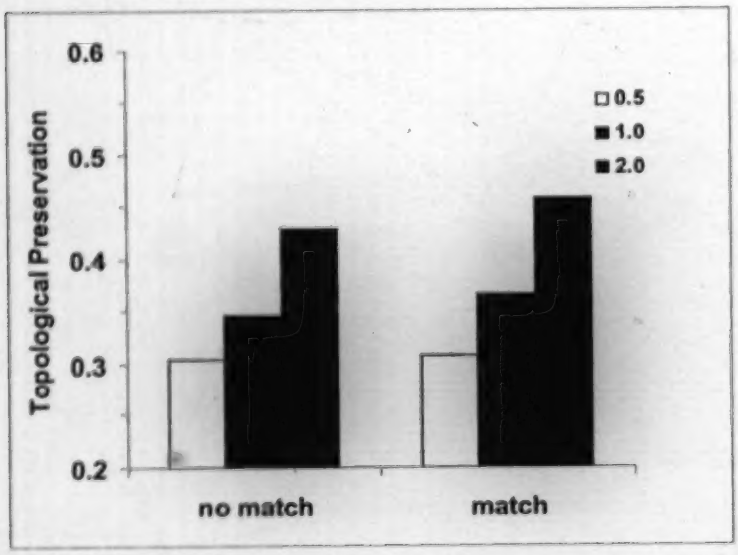


Figure 7.8. Mean Topological Preservation by Item Discrimination and Dimensional Match for All Large Map Conditions.



Correlation of Distances

An ANOVA using the same predictors as above was conducted to determine their respective roles in the prediction of R_{dist}. Table 7.13 shows the cell means for each combination of factors, and Figures 7.9 and 7.10 display these means for small and large map conditions, respectively. As with TP, each of the interaction effects and main effects were statistically significant. The 3-way interaction of all factors ($F(2, 2388) = 5.3$, $p<0.005$), indicates that the difference between match and non-match conditions depends upon both map size and item discrimination. The interaction of match and item discrimination, $F(2, 2388) = 577.8$, $p<0.001$, shows that dimensional match leads to greater increases in R_{dist} when item discrimination is larger. The weak interaction between match and map size, $F(1, 2388) = 6.165$, $p<0.05$ shows that differences between match and non match conditions are greater for large maps, despite smaller cell means for all conditions. Last, dimensional match was the strongest main effect, $F(1, 2388) = 3313.8$, $p<0.001$, underscoring its importance in the accurate projection of test data.

Table 7.13. Mean Correlation of Distances Across All Conditions by Map Size, Item Discrimination, and Dimensional Match

		Item Discrimination		
Map Size		0.5	1.0	2.0
4 x 4	No Match	0.63	0.69	0.69
	Match	0.67	0.87	0.94
8 x 8	No Match	0.48	0.58	0.61
	Match	0.52	0.77	0.91

Figure 7.9. Mean Correlation of Distances by Item Discrimination and Dimensional Match for All Small Map Conditions.

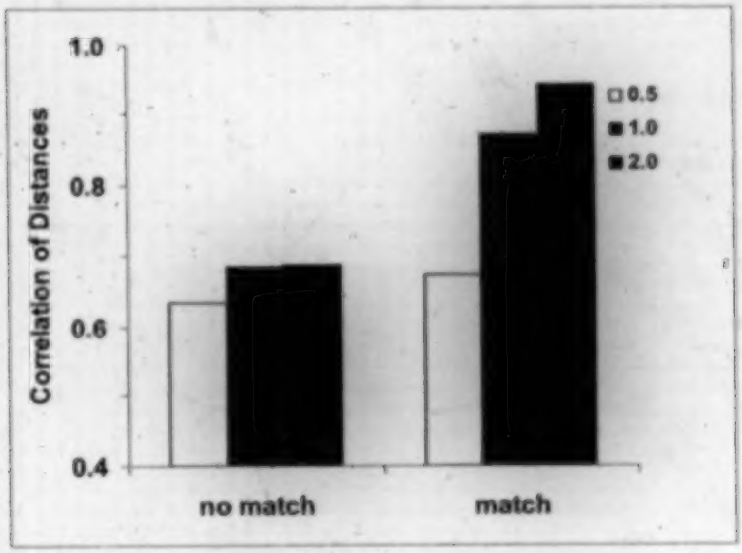
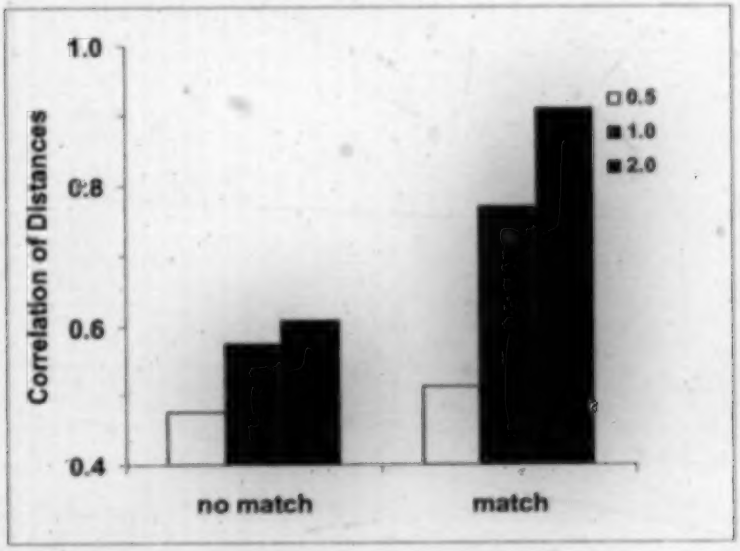


Figure 7.10. Mean Correlation of Distances by Item Discrimination and Dimensional Match for All Large Map Conditions.



Discussion – Combined Analysis

The results from the combined analysis are strong evidence for the importance of the dimensional match in using the SOM as a data projection method. The superiority of match relative to non-match conditions was most notably seen in the analysis of R_{dist} which reflected the SOM's capacity to preserve distance relationships from the test data. This analysis demonstrated superior projection across all conditions for maps whose dimensionality matched that of the data to those that did not.

Results from the analysis of Topological Preservation followed a similar pattern, but were more equivocal. The superiority of dimensional match was observed for conditions of item discrimination 1.0 and higher, but was not evident when item discrimination was equal to 0.5. This would seem to indicate a certain threshold of item quality in order to preserve adjacency relationships.

Last, Dimensional Match plays no apparent role in the minimization of Quantization Error. This finding is not unexpected. Since QE is equivalent to the expected error when using a model vector to represent a data point, only two factors should influence this measure: the number of model vectors and the complexity of the data. Since the number of model vectors was equal in each condition, the complexity of the data should entirely predict the variation in QE.

Chapter 8 – General Discussion

The preceding experiments examined the capability of Self-Organizing Maps to represent the structure of certain educational data. In particular, specific characteristics of the data, the maps, and the relationship between the two were varied to determine the effects on the statistical, qualitative, and interpretive characteristics of SOM representation of test structure. Certain conclusions about the capacity of SOMs in this regard can now be made, and a number of interesting implications result for both the future use of SOMs in this context, and for future research. These will be discussed in the present chapter.

First, the effect on test structure representation with respect to the characteristics of the data and the SOM are discussed. Next, the role of dimensional match in rendering test structure is discussed both with respect to map-data fit and projection. The conception of the SOM as a statistical, possibly latent class model for complex data is discussed, as are the implications for optimal map training and design. In particular, the notion of a *sufficient* SOM will be considered in light of the present results and criteria applied to other test structure methods. Last, the application of the SOM to certain educational measurement problems will be reviewed, and analyzed with respect to its appropriate use.

Using the SOM to Determine Test Structure - Considerations

Characteristics of Data

SOMs represent test structure best when the items composing the test have high discrimination. This is evidenced by both lower QE and higher values on measures of projection for those maps whose data had higher values of discrimination. This is not

surprising since in the present research, item discrimination, or more precisely the lack thereof, was a surrogate for the presence of random error. This finding is therefore tantamount to saying the SOMs represent test structure better when there is less random error in the data, a tautology.

A more specific question is how much random error SOMs can tolerate before the representation of test structure becomes significantly degraded. Though no specific criterion was used in the present research, highly visually interpretable maps were obtained in most conditions with item discrimination = 2.0. Interpretable maps, though often significantly compromised, were also observed at discrimination levels of 1.0, while levels of 0.5 generally produced maps with little evident structure.

It may be, however, that a more important variable for the generation of visually interpretable SOMs may be the *reliability* of the test as a whole rather than the discrimination of the individual items. Reliability is a measure of how consistently individuals having the same ability will receive similar test scores. In the SOM context, receiving a similar test score could be considered equivalent to the classification or assignment of individual observations to the same receptive field, or cluster (e.g., deBodt, Cottrell, & Verleysen, 2002). Since individual data were simulated as belonging to one of a finite number of classes, the role of reliability could be addressed by determining how consistently observations derived from the same latent class ended in the same cluster. Using similar logic, an alternative measure of this reliability is the homogeneity of class composition of each cluster. In Experiment One, it was shown that this homogeneity decreased with decreases in item discrimination and therefore some support for the role of reliability was demonstrated. As further evidence, a traditional measure of

test reliability could be calculated for the test data and compared with the interpretability of the SOM solutions. KR20 (Kuder & Richardson, 1937), one such index of reliability was calculated for each unidimensional dataset (i.e., the simulated test data used in Experiment One and Experiment Three) and is displayed in Table 8.1.

Table 8.1. KR-20 reliabilities for each unidimensional dataset.

	Item Discrimination		
	0.5	1.0	2.0
KR20 Reliability	0.569	0.798	0.885

The pattern shown in Table 8.1 appears to match the interpretability of SOMs across the three levels of item discrimination, across conditions. Therefore, test reliability and not necessarily item discrimination may be at the root of the differences in map interpretability. Since increasing the number of similar items on a test will increase the reliability for that test even when the discrimination of individual items is low (e.g., Spearman-Brown Prophecy Formula), a future experiment could compare SOMs from tests with different item discriminations but the same test reliabilities. Neither KR20 nor the Spearman-Brown formula is likely to perfectly anticipate improved performance of the SOM since these conceive of reliability only in terms of the consistency of the *total score* while the SOM will classify examinees based on Euclidian distances and the location of units in the map. However, the *principle* of reliability could help account for how well SOMs represent the structure of educational data.

Characteristics of the SOM

In general, smaller maps created better projections of the data structure, while larger maps had lower quantization error. The increases in projection for small maps does not imply that, in general, 'smallness' is a pre-requisite for good projection, rather that smaller maps seemed to be better suited for representing these particular data. Since the data comprised 4 latent classes when one-dimensional and 16 when two-dimensional, it is reasonable to assume that the large map comprising 64 units was not a natural medium for the projection of these data. Consequently, when item discrimination was high, the map appeared to attain some kind of order overall, but many locations in the SOM had receptive fields with no data. This resulted in lower values of topological preservation as compared with smaller maps because no adjacency relationships exist when adjacent fields are empty. When item discrimination was low, the projected data did not represent well the simulated structure. Though examination of typical large maps from each experiment (i.e., Figures 5.4, 6.3, 7.2, and 7.4) might suggest that the SOM is limited in its ability to order data appropriately, it is more likely that the data were ordered well enough, but the relationship of that ordering to the simulated class was degraded by the low discrimination. That is, the low item discriminations induced so much overlap between class distributions that many regions in the metric space did not reliably represent any particular class. In this case, the poor ordering with respect to simulated class apparent in the SOM is consistent with the properties of the data.

In general, the poorer representation of test structure from the use of large maps seems to result from the *over-fitting* of the data. That is, the large number of model vectors coupled with the relative small sample size means that some receptive fields will

be tuned to irrelevant characteristics of the data, insofar as the overall structure is concerned. This is an especially acute problem when there are a lot of 'irrelevant characteristics' in the data, such as would be the case when item discriminations are low. From this perspective, lower Quantization Error observed in the large map conditions may not reflect a higher quality of representation of the test data. Rather, each model vector is representing smaller regions of the data overall but these small regions are representing spurious characteristics of the data.

This over-fitting problem could be partially mitigated in several ways. First, model vectors would be more likely to represent 'true' characteristics of the class distributions if the sample size was increased. This would limit the effect of over-fitting by ensuring that certain receptive fields did not emerge strictly by an under- or over-representation of certain regions of the class distributions. Second, by increasing the above-mentioned reliability of the test data, the systematic characteristics of the data would become more salient and thus the SOM is more likely to becoming organized around those characteristics. Third, the selection of map size could be based on certain statistical criteria, such as those presented by deBodt, Cottrell, and Verleysen (2002), explained below.

deBodt and colleagues (2002) presented a method by which a certain type of reliability of classification is evaluated across replications of a SOM. In particular, they were interested in how consistently pairs of observations were adjacent in multiple runs of a SOM and also, how this consistency was affected by varying map size. When specific pairs of observations were consistently adjacent, it was argued that this adjacency represented true characteristics of the data. Furthermore, it was demonstrated that above

a certain map size, the consistency of adjacency relationships drops rapidly, much like 'elbows' are observed in scree plots of eigenvalues in a principal components analysis. deBodt, Cottrell, and Verleysen argued that the map size immediately preceding this drop should be adopted as the appropriate size, as it represents the best trade-off between the complexity of the data and over-fitting. This type of method could be fruitfully applied to educational data much like those presented in this thesis and is a promising topic of further research.

Dimensional Match between Data and Map

One of the foci of the preceding experiments was the role of a match in dimensionality between the SOM and the data in accurately identifying the dimensional structure of test data. The results demonstrated that this match was a necessary precondition for high quality projection, both from the perspective of topological preservation and correlation between distances in the map and metric spaces. This can be seen in Figures 7.7 through 7.10 where higher values were almost universally obtained for both of the above measures in conditions of dimensional match. This result was corroborated by the finding that from the qualitative perspective, maps generated in match conditions were more interpretable in terms of their dimensional structure and the classes that composed them. On the other hand, dimensional match appeared to have no significant impact on the quantization error in the map. Instead, maps comprising the same number of units had highly similar quantization error, irrespective of the dimensional structure of the map. This was seen most clearly in the Figures 7.5 and 7.6.

What these findings seem to indicate is that dimensional match does not significantly affect the *locations* of the model vectors in multidimensional space; only the

arrangement of these vectors in the SOM is influenced. Since QE is a measure of how well the model vectors represent the probability density of the space, no differences in QE between SOMs having the same number of units suggest that the model vectors in both represent the density equally well, despite the difference in the dimensionality of the map. If this assertion is true, then other properties of the sets of model vectors, irrespective of the dimensionality of the map from which they were generated, ought not to give away the dimensionality of the map. One such property is the dimensionality of the model vectors as revealed by MDS analysis. Comparing Table 5.7 with 7.4, one-dimensional data, and Table 6.4 with 7.9, two dimensional data, no systematic differences in Stress1 values were observed; in the one-dimensional case, the match conditions had lower Stress1 values 7 times out of 12 (58%), and in the two-dimensional case, the match conditions had lower values 5 of 12 times (42%). Though this was a small sample, the results do not suggest any role for dimensional match in the placement of model vectors and therefore, does not suggest any advantage of dimensional match for quantization error, either.

The selective improvement in projection but not quantization error for conditions of dimensional match has several implications for using SOMs to determine test structure. First, the SOM ought to be considered to reflect test structure only when the dimensionality of the map and data are the same. This could be determined in much the way as presented here; different SOMs with varying number of dimensions could be used to analyze the same data. The SOM chosen would be that with the highest values of TP and R_{dist}. Complicating matters is the fact that dimensional match is not the sole determinant of projection; map size plays an important role. Determining the best

combination of dimensions and map size could be a kind of titration process, ending when values for projection begin decreasing with further systematic changes in parameters. A second implication concerns the use of the SOM as a tool to reveal qualitative properties of the data. Using adjacency and proximity in the SOM as an analogy to the same properties in the test data are only recommended when TP and R_{dist} are maximized, that is, under conditions of dimensional match. Later in this chapter, a specific application of a SOM in an educational measurement context will be evaluated with respect to this criterion. Last, an optimally-projected map is not the only source of test structure information provided by the SOM. The model vectors themselves contain information about the underlying dimensional structure of the data that can be extracted using other analytical techniques such as MDS. Adopting well-established criteria regarding the use of MDS in determining the number of dimensions could be applied (e.g., Kruskal & Wish's [1978] criterion). This procedure would have the advantage of bypassing the need to determine the dimensionality of the SOM in parallel with the dimensionality of the data. Instead, a SOM comprising any number of dimensions could be generated and the dimensionality implied by the MDS analysis could then inform the dimensionality of the SOM to create the optimal projection of the data. What remains to be determined is how robust each of these processes (i.e., projection- and QE-based approaches to determination of dimensionality) is with respect to variables important for educational measurement. These variables include correlation between dimensions, simple versus complex structure, strength of secondary dimensions, and number of items representing these dimensions. Of course, this is an area for future research.

SOMs as statistical models

Until this point, the capacity of SOMs to determine the overall dimensional structure of data has been the primary focus. What remains to be addressed are the conditions under which SOMs produce *optimal* representations of the data with which they are trained. From the data presented here it is clear that map size plays a significant role in the *clarity* with which the map reveals the underlying dimensional structure and therefore, a critical examination of the role of map size could have important implications for creating such optimal representations. Interestingly, other statistical methods and criteria used in educational measurement may have important implications for addressing this problem. These methods and their implications are explored in the next section.

As mentioned above, the optimal number of units in the map relates to the complexity of the data overall. Recall that Stevens et al. (1999) used complexity of test data as a justification for increases in map size. To this author's knowledge, clear rationale for the choice of map size has largely been missing from applications of SOMs in educational measurement. Even deBodt, Cottrell, and Verleysen (2002) used only square maps in evaluating appropriate size. In the two real-data examples they present, 6 x 6 and 7 x 7 maps were chosen as best representing their data, an increase of 11 and 13 units over the next best-fitting maps, respectively. Though these map parameters were deemed *necessary* for good fit of the data, little or no emphasis was placed on determining the *sufficiency* of these maps for the data they represent. As an example, maybe only 5 additional units were sufficient to achieve optimal fit rather than 11 or 13.

In the present study, the number of latent class distributions in the data could be considered an index of data complexity. Since the number of latent classes was never larger than the number of units and it was consistently found that smaller maps better

represented the data in terms of projection, it was argued in a previous section that the smaller map conditions better match the complexity of the data. But how many units might be sufficient to represent those data? For instance, could a single unit adequately represent a class *distribution*? One source of support for this notion is that the 'unit conditional' item probabilities were obtained from the SOM (see Chapter 5, p. 19), and these could relate directly to the class conditional item probabilities featured in most latent class models. Since a single set of class conditional item probabilities combined with the class probability defines the entire set of parameters necessary to estimate a latent class model, it may appear that an integration of class (unit) prevalence in the SOM is all that is needed to convert a SOM into a latent class model of the type pioneered by Lazarsfeld and Henry (1968). However, there are several important differences between units in the SOM and the classes as defined in LCA. First, in LCA individual observations are assigned to latent classes based on a Bayesian classification rule; the SOM makes this assignment based on Euclidian distance. The Bayesian rule features two components that determine class membership, *likelihood* and *prior probability*. The first component, likelihood, determines a probabilistic weight that individual observations 'belong' to each class. This is determined by the product of individual class conditional item probabilities for a given observation (see Chapter 3, p. 13). This is similar to the calculation of proximity in the SOM with one important difference; the calculation of likelihood in LCA takes into account the unique variance of the class conditional distributions in multiple directions. Using Euclidian distance for all class assignments implies that the variance in each direction of each component distribution is the same. The second component of the Bayesian rule is an explicit modeling of the prior

probability of belonging to each of the latent classes. SOMs model only the probability density of regions in the composite distribution.

Because of these properties of the SOM, Kohonen (2001) asserts that the SOM must have a many to one, units to class distributions relationship in order to optimally represent the topology, or latent structure of the data. Furthermore, unlike the Bayesian framework in LCA, the SOM itself is not meant to be a classification tool; it is a method to reveal overall structure in data. To optimize the classification function of the map, Kohonen argues that *supervised* training known as Learning Vector Quantization (LVQ) is necessary. The result of LVQ is to position the model vectors so that the resulting decision boundaries best *approximate* the Bayesian, or optimal boundary.

In principle, however, there is nothing preventing the importation of components of Bayesian classification for use in the SOM. In particular, the calculation of distance used by LCA models could be used as the training rule for SOMs. Moreover, the prior probability of each class could also be explicitly included in the training formula, weighting the likelihood of classification of individual patterns. If successful, SOMs could be used as an additional computational method to creating latent class accounts of test structure. This method could be considered an Expectation / Maximization (EM) approach and bears close similarity to K-means cluster analysis, also an EM method.

Why would a SOM-based method of latent class analysis present an advantage for analyzing educational data? First, the projection capabilities of the SOM could augment an account of test structure derived from LCA. That is, the ordered representation of latent classes generated by the SOM could provide a layer of interpretation not currently available in LCA models. More fundamentally, this method could represent a means of

representing student performance both in terms of dimensional and latent class structure. That is, student performance could be viewed, not only from the perspective of the dominant abilities underlying performance on the test and therefore the possession of a hypothetical latent trait, but also in terms of specific patterns of correct and incorrect responses and what those patterns might imply about the states of mastery for particular examinees. From the results discussed above, it is clear not only that a dimension-based account of test structure is possible from the SOM, but also what are some of the features essential to optimizing that account. Creating a latent class account of test structure using SOMs, to this author's knowledge, has yet to be systematically investigated. However, from experience with other latent class methods such as cluster analysis and LCA, some of the essential criteria of such an account are already known. Primary among these is a means to determine the sufficiency of the latent class model for representing the data. The variant of LCA pioneered by Lazarsfeld and Henry (1968) uses the criteria of conditional independence of item responses to determine the value of latent class parameters. Those parameters could then be used to determine how well the values of the latent parameters reconstruct the observed data (see Chapter 3, p. 15). These same criteria could be applied to determine the optimal number of units for a SOM, with one important caveat; because they are empirically- and not parametrically-defined, the values of so-called conditional probabilities in the SOM are themselves observed and not latent variables. This implies that, instead of determining how well latent parameters for distributions are able to reconstruct observed data such as is the case in LCA, the SOM approach would require an examination of the statistical properties of the observed conditional probabilities. That is, in order to determine if a SOM is 'sufficient' to

represent a set of data from the latent class perspective, the unit conditional probabilities would be evaluated with respect to statistical criteria, such as conditional independence. This could be accomplished in much the same way as is done in the DETECT procedure (see Chapter 2, p. 14); the correlations between performance on items could be calculated separately for each unit, and then aggregated together noting the strengths of these conditional relationships. When the number of units is sufficient to achieve conditional independence to a specific criterion, (see, for example, Stout, 1987), the SOM variant of LCA could be considered sufficient.

The uniqueness of the SOM notwithstanding, using implicit or explicit latent class models as a conditioning variable is common procedure in educational measurement. In fact, determining whether data meets a criterion of conditional independence is *predicated* on the definition of latent classes in the data. Like the DETECT procedure, item response theory (IRT) models use implicit definitions of latent classes to determine the dimensionality of the assessment. For these models, a latent class is defined by the total number of items answered correctly; all examinees that answer the same number of items correctly constitute that class. When statistical independence is achieved within each such group, (essential) unidimensionality is concluded. Moreover, the use of latent classes to define IRT models is not limited to the determination of dimensionality. In particular, latent class models with ordinal constraints have been used to define the parameters of IRT models themselves (e.g., Croon, 1990, 1991; Lindsay, Clogg, & Grego, 1991; Vermunt, 2001). These models and SOMs both make explicit that strictly trait-based accounts of performance are fundamentally related to class-based accounts.

This acknowledgement lends further support to the exploration of SOMs in the representation of test data.

The fundamental difference between the use of implicit latent classes in educational measurement (such as those used in IRT models) and latent class derived from a SOM is in the treatment of class-level information. In the IRT context, latent classes are used as tools to determine dimensionality, and the specific characteristic of interest is the correlation or covariance between items. In the SOM context, the class could become a different source of information, one that reveals general characteristics of the response patterns elicited from members of each class. This type of class could still be examined for dimensionality in the same manner as total score based latent classes, but could also be used to create more *substantively* based descriptions of test performance. Just as in LCA, class conditional item probabilities could be examined to determine 'typical' patterns of performance for subgroups on the test defined not by the number of items they answered correctly, but by the specific patterns of correct and incorrect responses on the test. This same type of information has been used in numerous *diagnostic* models, with considerable success (e.g., Leighton, Gierl, & Hunka, 2004; Mislevy, 1996; Tatsuoka, 1990, 1995). Further investigation of LCA-based SOMs could help more clearly determine the potential of SOMs for educational measurement in this regard.

In summary, an implication of the present research for applications of the SOM in educational measurement is that it provides a latent class account of test performance. These classes could serve as the basis for certain dimensionality assessment procedures in which a conditioning variable is necessary. A further enticing application is the use of the latent class structure to develop a substantive interpretation of the test's latent space.

This kind of interpretation becomes possible when the patterns of correct and incorrect responses implied by the unit conditional item probabilities are interpretable in terms of test content. Examining this possibility is a priority for future research.

An Evaluation of a Current Application of SOMs in Educational Measurement

The present research has several important implications for the appropriate use of SOMs in educational measurement. One, in order to maximize the *meaningfulness* of the SOM topology for interpretation of the data, the intrinsic dimensionality of data and the number of dimensions of the SOM must match. When this characteristic does not hold adjacency and distance relationships between the map and metric space are compromised, leading to potential errors in interpretation. Two, map size plays a key role in determining the quality of representation of the data in the SOM. Three, item discrimination, or perhaps test reliability, is an important variable in generating well-fitting and interpretable maps. These three characteristics shall be used to evaluate a recent application of a SOM in educational measurement, Stevens, Johnson, and Soller (2005).

The recent research by Stevens, Johnson, and Soller (2005) continues in the theme of using SOMs to elucidate performance on complex tasks, an approach Stevens and his colleagues have pioneered. The basic approach follows that of Stevens et al. (1999) outlined in Chapter 4, p. 69. In the present case, SOMs were used for the, "categorization of common strategies by artificial neural network clustering (p. 44)" for several genetics simulations. The particular purpose of the SOMs in this research was to inform the identification of performance states and to later model using Markov processes the trajectory through those states that come with increases in competence. The network

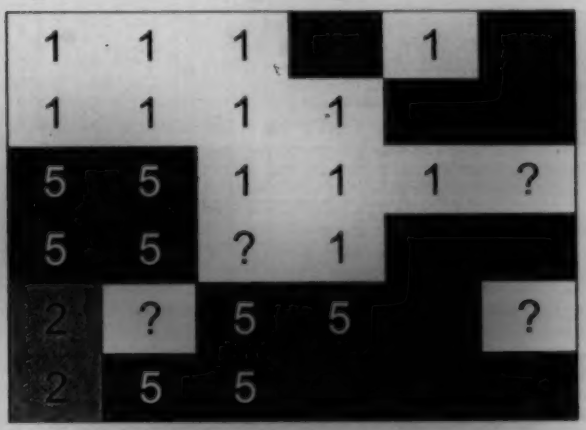
chosen to accomplish this was 2-dimensional, typical of Stevens' previous work. They describe the functioning of SOMs as follows, "The mathematics behind self-organizing neural networks is such that groups of similar performances appear on an output 6 x 6 grid of classifications as physically near each other (Kohonen, 2001). ANNs yield a 'topological map' of similar performances in which the geometric distance between nodes is a metaphor for similar solving strategies (p. 45-6)".

No specific information is provided regarding the intrinsic dimensionality of the data, and therefore, it is not known whether the distance in the 'topological map' will, in fact, be a good metaphor for performance similarity. A figure is provided that helps address this question, a representation of which is presented below as Figure 8.1. In this figure, five performance states are identified, presented as increasing in sophistication from state 1. In Chapter 5, two criteria were used to evaluate the qualitative representation of the data by the SOM, the contiguity of regions representing the same class, and the smoothness of ordering of the classes across the projected space. These criteria are applied to the representation of performance states in Stevens', Johnson, and Soller (2005).

The map shows limited contiguity of the states portrayed in the map. Contiguous regions exist for some states and not others. Only one state has all its units together (State 2) but it only has two units in the SOM. Smoothness of ordering of classes in the SOM would reflect the importance of Euclidian distance in defining similarity of performance quality. Clearly, this distance played a limited role in determining the competence of individuals working on these tasks since ordering was not evident in the

map. Based on these criteria, it is questionable whether the ordering property of SOMs is insightful into the quality of task performance in this research.

Figure 8.1. Representation of performance states derived from self-organizing neural network in Stevens, Johnson, & Soller (2005).



An interpretation of the map is offered in the article, but not with respect to performance quality. Rather, map location seemed to be more closely linked to the selection of particular pieces of information. For example, they indicate that a certain region of the map, "…is where students select a large number of items, but no longer use the Glossary. These strategies are represented on the right-hand side [of the map] (p.46)". Other regions, "… illustrate another qualitative difference in which there is predominantly a usage of enzyme assays (p.46)". Similarities between strategies and not either the overall dimensional structure or the relationship of the SOM representation to performance quality appeared most important in their analysis.

Based on this analysis and the results from this thesis, several conclusions can be made about the appropriateness of the use of SOMs in the work by Stevens and his

colleagues. First, it was not apparent that the characteristics identified as important by the present research were embodied in their work. In particular, no identification of the dimensionality of the responses data was made and no particular justification was offered for the selection of a 6 x 6 map. This does not in itself invalidate the use of the SOM in this context, as the purpose of its use was not creating dimensionally-based interpretations of test structure. That similar strategies occupied similar regions in the SOM appeared to provide some helpful information in determining the performance states underlying the task. However, the fact that performances classified as similar in quality did not always occupy similar regions in the map undermines the utility of the SOM in determining these states. Instead, it appeared that substantive considerations trumped the analysis of similarity performed by the SOM. The choice of a 6 x 6 map to represent these data seemed to be somewhat arbitrary. Since it was discovered in the present thesis that map size plays a significant role in the representation of the structure, it could be that different states would be identified by maps of different size. Last, the importance of item discrimination or test reliability did not seem to readily apply to this context. That is, since the map in this research was organized around the selection of particular pieces of information in the performance tasks and not in responses to test items, measures of the quality of test items, that is item discrimination, are not transparently relevant. Therefore, little insight into the importance of this variable for Stevens' work can be gained. However, the discovery of a consistent trajectory through the performance states in this research indicates a relationship between competency and the type of strategy employed. This could be interpreted as evidence for the relevance, if not reliability of the data collected for the classification generated.

Conclusion

The research presented in this thesis demonstrates that, when used appropriately, SOMs are capable of revealing the structure of test data. A clear understanding of the boundary conditions under which it is capable of doing so, however, is a question for future research, as is its performance relative to existing methods for determining test structure. Important and potentially exciting applications for SOMs appear to reside in creating latent class accounts of test performance. Though current applications of SOMs in educational measurement focus on their use in alternative assessments such as complex performance tasks and automated scoring, the present research points to future potential of the method in more mainstream applications. Should this potential be realized, the application of Kohonen Self-Organizing Maps in educational measurement may become more extensive and wide-ranging. Should that be the case, it is hoped that this thesis, and research that follows from it, will have helped to provide insight in determining its effective utilization.

References

Croon, M.A. (1990). Latent class analysis with ordered latent classes. *British Journal of Mathematical and Statistical Psychology, 43*, 171-192.

Croon, M.A. (1991). Investigating Mokken scalability of dichotomous items by means of ordinal latent class analysis. *British Journal of Mathematical and Statistical Psychology, 44*, 315-331.

de Bodt, E., Cottrell, M., & Verleysen, M. (2002). Statistical tools to assess the reliability of self-organizing maps, *Neural Networks, 15*, 967-978.

Kohonen, T. (2001). *Self-organizing maps* (3rd ed.). Berlin: Springer-Verlag.

Kruskal, J. B., & Wish. M. (1978). *Multidimensional Scaling.* Sage Publications. Beverly Hills. CA.

Kuder, G. F., & Richardson, M. W. (1937). The theory of estimation of test reliability. *Psychometrika. 2,* 151-160.

Lazarsfeld, P. F. & Henry, N. W. (1968). *Latent structure analysis.* Boston: Houghton Mifflin.

Leighton, J. P., Gierl, M. J., & Hunka, S. (2004). The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach. *Journal of Educational Measurement, 41,* 205-236.

Lindsay, B., Clogg, C. C., & Grego, J. (1991). Semiparametric estimation in the Rasch model and related exponential response models, including a simple latent class model for item analysis. *Journal of the American Statistical Association, 86*, 96-107.

Mislevy, R. J. (1996). Test theory reconceived. *Journal of Educational Measurement, 33*, 379-416.

Stevens, R., Ikeda, J., Casillas, A., Palacio-Cayetano, J., & Clyman, S. (1999). Artificial neural network-based performance assessments. *Computers in Human Behavior, 15*, 295-313.

Stevens, R., Johnson, D., & Soller, A. (2005). *Cell Biology Education, 4*, 42–57.

Stout, W. (1987). A nonparametric approach for assessing latent trait unidimensionality. *Psychometrika, 52*, 589-617.

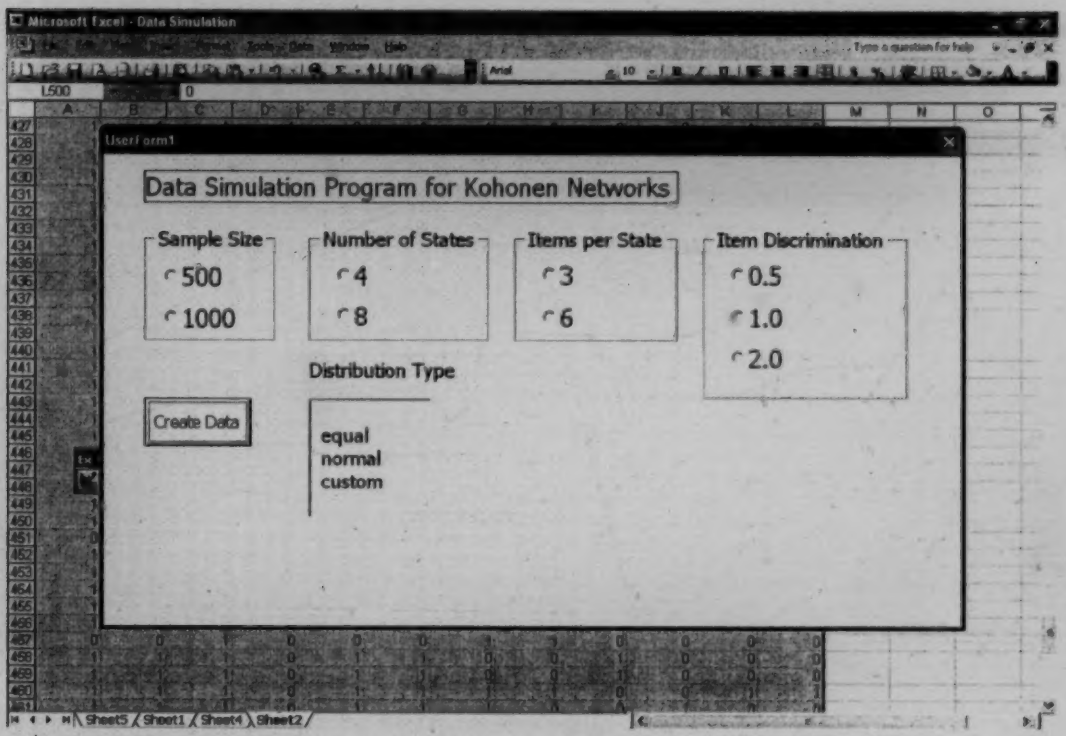
Tatsuoka, K. K. (1990). Toward an integration of item-response theory and cognitive error diagnoses. In N. Fredericksen, R. L. Glaser, A. M. Lesgold, & M. G. Shafto (Eds), *Diagnostic Monitoring of Skill and Knowledge Acquisition*. Hillsdale, NJ: Erlbaum.

Tatsuoka, K. K. (1995). Architecture of knowledge structures and cognitive diagnosis: A statistical pattern recognition and classification approach. In P. Nichols, S. F. Chipman, & P. L. Brennan (Eds.), *Cognitively Diagnostic Assessment* (pp. 327-359), Hillsdale, NJ: Erlbaum.

Vermunt, J. K. (2001). The use of restricted latent class models for defining and testing nonparametric and parametric item response theory models. *Applied Psychological Measurement, 25*, 283-294.

Appendix A – Data Generator: User Form and Visual Basic Code

Figure A1. Form to Specify Data Generation Parameters



Note: Not all settings were used for the present research.

Visual Basic Code

Code for Form Controls

```
Private Sub CreateData_Click()        'All subs in Data_Generator module

    Initialize
    GetAbility
    Quantize
    SetItemParameters
    GenerateItemResponses
    UserForm1.Hide
    Sheet2.Activate
    UserForm2.Show

End Sub
```

```
Private Sub Sample500_Click()

 Current.SampleSize = 500

End Sub
```

```
Private Sub Sample1000_Click()

 Current.SampleSize = 1000

End Sub
```

```
Private Sub FourStates_Click()

Current.NumberOfStates = 4

End Sub
```

```
Private Sub EightStates_Click()

Current.NumberOfStates = 8

End Sub
```

```
Private Sub ThreeItems_Click()

Current.ItemsPerState = 3

End Sub
```

```
Private Sub SixItems_Click()

Current.ItemsPerState = 6

End Sub
```

```
Private Sub Title_Click()

MsgBox (Current.SampleSize)

End Sub

---

Private Sub ItemDiscrimPoint5_Click()

Current.ItemDiscrimination = 0.5

End Sub

---

Private Sub ItemDiscrimOne_Click()

Current.ItemDiscrimination = 1.0

End Sub

---

Private Sub ItemDiscrimTwo_Click()

Current.ItemDiscrimination = 2.0

End Sub

---

Private Sub UserForm_Activate()      'Populate Listbox in UserForm1

Dim ListArray(4) As String

    ListArray(1) = "equal"
    ListArray(2) = "normal"
    ListArray(3) = "custom"

    DistList.List() = ListArray

End Sub

---

Private Sub DistList_Click()

Dim j As Integer                     'loop counter
Dim z() As Double                    'holds z-values for conversion to probabilities

    Select Case DistList.ListIndex

        Case 1
            ReDim Current.Cutpoints(Current.NumberOfStates)
            If Current.NumberOfStates = Null Then GoTo EnterValues
            For j = 1 To Current.NumberOfStates
                Current.Cutpoints(j) = j * Current.SampleSize / Current.NumberOfStates

         'Divides examinees into equal size classes

            Next j

        Case 2
            ReDim Current.Cutpoints(Current.NumberOfStates)
            ReDim z(Current.NumberOfStates)

            If Current.NumberOfStates = Null Then GoTo EnterValues

            For j = 1 To Current.NumberOfStates
                z(j) = (-3 + (j * 6 / Current.NumberOfStates))
```

```
            If z(j) = 3 Then
                Current.Cutpoints(j) = Current.SampleSize
            Else
                Current.Cutpoints(j) = Current.SampleSize * (1/(1+Exp(-1.7 * z(j))))
            End If

            'divides distribution (-3, 3) into chunks of equal standard score size,
            'based on logistic approximation to cumulative normal distribution

        Next j

    Case 3      'file will manually specify values for all parameters

        inputfile = InputBox("Enter filename")
        Open inputfile For Input As #1

        Input #1, Current.NumberOfStates
        Input #1, Current.SampleSize
        Input #1, Current.ItemsPerState

        ReDim Current.Cutpoints(Current.NumberOfStates)
        For j = 1 To Current.NumberOfStates
            Input #1, Current.Cutpoints(j)
        Next j

        Close #1

End Select

Exit Sub

EnterValues:
    MsgBox ("Enter Number of States first.")

End Sub
```

Data_Generator module

```
Option Explicit

Type Data                               'Characteristics of simulated data

    SampleSize As Integer               'number of simulees
    NumberOfStates As Integer           'number of knowledge states
    StateDifficulty() As Double         'the 'b' value of each knowledge state
    ItemsPerState As Integer            'number of items for each above state
                                        'NOTE: NumberOfStates * ItemsPerState = Number of
                                            Items
    ItemDiscrimination As Double        'the 'a' parameter  NOTE: This could be an array
                                            for more detailed sims
    ItemDifficulty() As Double          'the 'b' parameter
    Responses() As Integer              'simulee response data
    Cutpoints() As Integer              'the index of 1st examinee in new group

End Type

Public Current As Data                  'the active simulation
Public SimuleeResponses() As Integer    'the observed responses by examinee

Public Const regular As Integer = 1     'regular random number
Public Const gaussian As Integer = 12   'normally distributed random number
```

```
'SUBROUTINES

Public Sub Initialize()                    'Clear all data from Worksheets

    Sheet1.Cells.ClearContents
    Sheet2.Cells.ClearContents
    Sheet1.Activate

End Sub
```

```
Public Sub GetAbility()                    'This sub creates a standardized, normally
                                           'distributed dataset in order to provide b-
                                           'parameter values for items and theta-
                                           'parameter values for class ability

Dim i As Integer                           'loop counter
Dim sum, mean, sd As Double

    For i = 1 To Current.SampleSize

        Sheet1.Range("simdata").Cells(i).Value = GetRandom(gaussian)

        'returns random value from an approximately gaussian distribution
        'and puts in column 1 of Sheet1

        'NOTE: simdata defines a data range on Sheet1: Column 1

    Next i

    mean = WorksheetFunction.Average(Sheet1.Range(Cells(1, 1), Cells(Current.SampleSize, 1)))
    sd = WorksheetFunction.StDev(Range(Cells(1, 1), Cells(Current.SampleSize, 1)))

    For i = 1 To Current.SampleSize

        Cells(i, "A").Value = (Cells(i, "A").Value - mean) / sd

        'standardizes values

    Next i

    Range("simdata").Sort (Columns("A"))

End Sub

Public Function GetRandom(TypeOfData As Integer) As Double

Dim result, num, sum As Double
Dim i As Integer

Select Case TypeOfData                     'gets a random number
    Case regular
        result = Rnd()

    Case gaussian                          'Creates approximately normally distributed data
        For i = 1 To 12                    'by summing random values
            Randomize
            num = Rnd()
            sum = sum + num
        Next i
        result = sum

    End Select

GetRandom = result

End Function
```

```
Public Sub Quantise()                    'Sets all theta values within a class

Dim i, j, n, index As Integer

ReDim Current.StateDifficulty(Current.NumberOfStates)

        For j = 1 To Current.NumberOfStates

                Current.StateDifficulty(j) = Range("simdata").Cells(Int((index +
                        Current.Cutpoints(j)) / 2)).Value

    'Sets the category score to the theta value of the simulee
    'in the middle of the category

                For i = index + 1 To Current.Cutpoints(j)
                    Range("simdata").Cells(i).Value = Current.StateDifficulty(j)
                Next i

                    index = Current.Cutpoints(j)

        Next j

End Sub
```

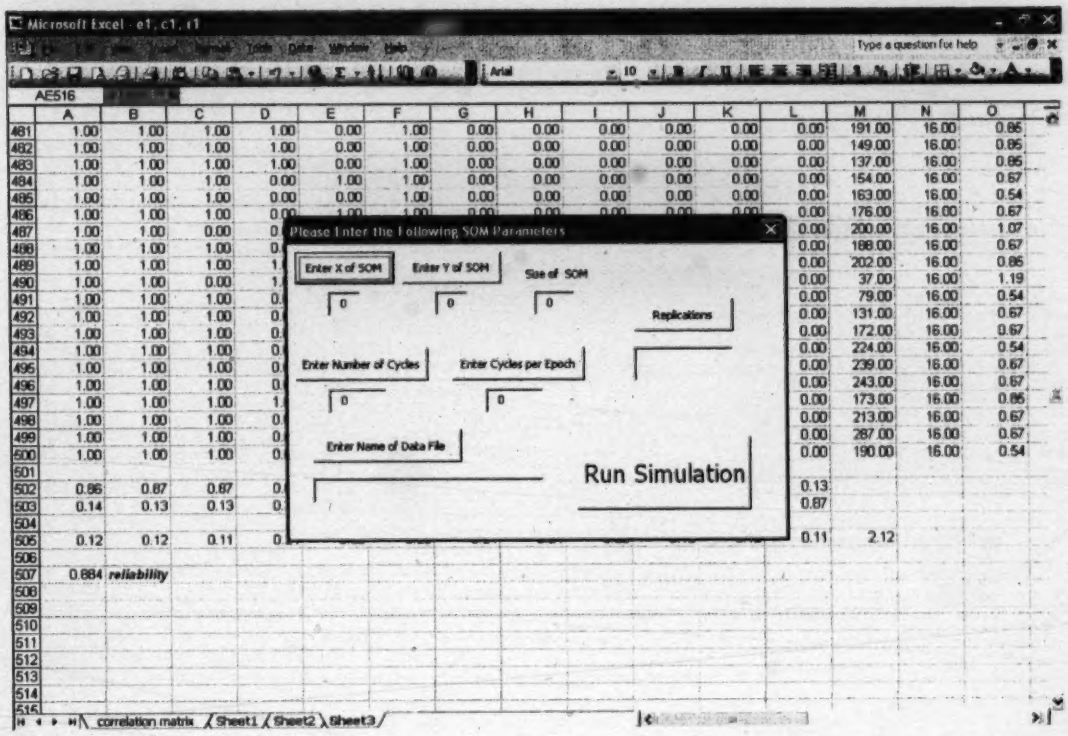
```
Public Sub SetItemParameters()

Dim i As Integer
Dim NumberOfItems As Integer

    NumberOfItems = Current.NumberOfStates * Current.ItemsPerState

    ReDim Current.ItemDifficulty(NumberOfItems)

    For i = 1 To NumberOfItems
        Current.ItemDifficulty(i) = Current.StateDifficulty(1 + Int((i - 1) /
                    Current.ItemsPerState))

        'Assigns difficulty value to the item based on item number,
        'number of states, and number of items per state.

    Next i

End Sub
```

```
Public Sub GenerateItemResponses()

Dim i, j, NumberOfItems As Integer

NumberOfItems = Current.NumberOfStates * Current.ItemsPerState

ReDim Current.Responses(Current.SampleSize, NumberOfItems)

    For i = 1 To Current.SampleSize
        For j = 1 To NumberOfItems

            Current.Responses(i, j) = 1
            If (GetRandom(regular) > IrtProb(i, j)) Then Current.Responses(i, j) = 0
            Sheet2.Cells(i, j).Value = Current.Responses(i, j)

        'Compares random number between 0 and 1 to IRT based probability of correct
        'response.  If equal or greater, returns 'correct' for that item.

        Next j
    Next i

End Sub
```

```
Public Function IrtProb(ByVal simulee As Integer, ByVal item As Integer) As Double

Dim prob As Double

    prob = 1 / (1 + Exp(-1.7 * Current.ItemDiscrimination *
                    (Worksheets("Sheet1").Cells(simulee, 1).Value
                        - Current.ItemDifficulty(item))))

        'Using 2PL IRT model to generate probability that examinee at particular level
        'of theta would answer item correctly

IrtProb = prob

End Function
```

Appendix B – SOM Engine: User Form and Visual Basic Code

Figure B1. Form to Set SOM Parameters



Visual Basic Code

Code for Form Controls

```
Private Sub UserForm_Activate()        'Sets system variables to values in text boxes

XText.Value = run.XofMap
YText.Value = run.YofMap
SizeofMapText.Value = run.SizeOfMap
CyclesText.Value = run.cycles
EpochText.Value = run.EpochSize
FileText = run.FileStem

End Sub
```

```
Private Sub Replications_Click()       'Sets value for number of replications

simnum = InputBox("Enter number of replications: ")
ReplicationsText.Value = simnum

End Sub
```

```
Private Sub EnterCycles_Click()        'Sets value for number of cycles

run.cycles = InputBox("Enter number of cycles: ")
CyclesText.Value = run.cycles

End Sub
```

```
Private Sub EnterEpoch_Click()         'Sets value for number of cycles before updating
                                       'model vectors

run.EpochSize = InputBox("Enter Number of Cycles per Epoch: ")
EpochText.Value = run.EpochSize

End Sub
```

```
Private Sub EnterX_Click()             'size of x dimension of map

run.XofMap = InputBox("Enter the width of the map: ")
XText.Value = run.XofMap
run.SizeOfMap = run.XofMap * run.YofMap
SizeofMapText.Value = run.SizeOfMap

End Sub
```

```
Private Sub EnterY_Click()             'size of y dimension of map

run.YofMap = InputBox("Enter the height of the map: ")
YText.Value = run.YofMap
run.SizeOfMap = run.XofMap * run.YofMap
SizeofMapText.Value = run.SizeOfMap

End Sub
```

```
Private Sub FileName_Click()          'Sets value for name of file stem

FileText = InputBox("Enter Name of the Data File Stem: ")

End Sub
```

```
Private Sub RunSimulation_Click()     'What to do when user clicks the RUN SIMULATIONS
                                      'button

    Sheet1.Activate
    ReadInputFile (FileText & ".dat")

        For runs = 1 To simnum
            InitializeStuff
            RunSOM
            run.FileStem = FileText & runs
            CalculateMeasures
            SaveData
        Next runs

End Sub
```

SOM Engine Module

```
Public Const GATE = 1
Public Const GAUSSIAN = 2
Public Const Test = 0
Public Const PI = 3.1415927

Type Simulation                    'The SIMULATION class has properties of the
                                   'data, SOM, and simulation parameters.

    FileStem As String                 '-file handle
    SampleSize As Integer              '-number of simulated examinees in data
    NumberOfItems As Integer           '-number of dichotomous items in data
    SizeOfMap As Integer               '-number of units in the map (entered by user)
    XofMap As Integer                  '-width of map (entered by user)
    YofMap As Integer                  '-height of map (entered by user)
    cycles As Long                     '-number of iterations
    EpochSize As Integer               '-number of cycles before updating model vectors

End Type

Type unit                          'The UNIT class contains properties of each unit
                                   'of the SOM

    locationx As Integer               'X-coordinate in SOM
    locationy As Integer               'Y-coordinate in SOM
    vector() As Double                 'element of model vector (
    PatError() As Double               '(euclidian) difference between each element
                                       'of model vector and pattern
    CycleError() As Double             'total error at each element over
                                       'all patterns in an "epoch"

End Type

Public patterns() As Integer       'array storing current input pattern
Public run As Simulation           'current simulation parameters
Public runs As Integer             'number of replications
Public layer() As unit             'array of units in map
Public simnum As Integer           'index for replication number
Public target As Integer           'the index for the input data to be matched
Public winner As Integer           'the closest unit to pattern in metric space
Public second As Integer           'the second closest unit to pattern
Public SOMadj, TE As Integer       'are two units adjacent (1=yes, 0=no)
```

```
Public Sub ReadInputFile(file As String)

Dim i, j As Integer                'loop counters
Dim line As String                 'input data

Open file For Input As #1
Input #1, run.SampleSize
Input #1, run.NumberOfItems

ReDim patterns(run.SampleSize, run.NumberOfItems + 1)
                                   '+1 so that array can store index number as well
For i = 1 To run.SampleSize
    Input #1, line
    For j = 1 To run.NumberOfItems
        patterns(i, j) = Mid(line, j, 1)
                                   'dichotomously scored data fills "patterns" array
    Next j
    patterns(i, j) = i             'index stored so pattern can be identified
                                   'as belonging to a particular class
Next i

Close #1

End Sub
```

```
Public Sub InitializeStuff()

    Sheet1.Cells.ClearContents      'Clear Model Vectors Sheet
    Sheet2.Cells.ClearContents      'Clear Measures Sheet
    Sheet3.Cells.ClearContents      'Clear Data and Classification Sheet
    ModelsRandomInit                'Initialize Model Vectors to random locations
    ClearErrors                     'Initialize Error associated with each model vector
    SetLocations                    'Assigns SOM locations to units, based on co-ordinate
grid

End Sub
```

```
Public Sub ModelsRandomInit()

Dim l, v As Integer                 'loop counters

ReDim layer(run.SizeOfMap)

For l = 1 To run.SizeOfMap          'the number of units in SOM

        ReDim layer(l).CycleError(run.NumberOfItems)
        ReDim layer(l).PatError(run.NumberOfItems)
        ReDim layer(l).vector(run.NumberOfItems)

    For v = 1 To run.NumberOfItems  '-number of elements in each model vector
        Randomize                   '-seed the random number generator
        layer(l).vector(v) = Rnd    '-inserts random number between 0 and 0.999
    Next v                          '   for each element of each model vector
Next l

End Sub
```

```
Public Sub ClearErrors()            'initialize error for each unit in the map

Dim l, v As Integer

For l = 1 To run.SizeOfMap
    For v = 1 To run.NumberOfItems
        layer(l).PatError(v) = 0
    Next v
Next l

End Sub

Public Sub SetLocations()           'sets locations of each of the units in the map
                                    'based on a rectangular map (not hexagonal)
Dim x, y, i As Integer

i = 1
For x = 1 To run.XofMap
    For y = 1 To run.YofMap
        layer(i).locationx = x
        layer(i).locationy = y
        i = i + 1
    Next y
Next x

End Sub
```

```
Public Sub RunSOM()

Dim weight As Double

Dim l, v As Integer                    'loop counters
Dim t As Long                          'iteration number

TE = 0

'This Select... Case statement determines which input pattern to train with
'and also randomly orders all the input vectors (Scramble Patterns)

For t = 1 To run.cycles

    Select Case (t Mod run.SampleSize)
                                       'finds the index of the
                                       'current input pattern

        Case 0                         't must be a multiple of the total number of
                                       'patterns therefore, present pattern number "n"
            target = run.SampleSize

        Case 1                         'We are just beginning a new presentation of the
                                       'entire set of patterns

            ScramblePatterns           'randomize pattern order
            target = t Mod run.SampleSize
                                       'Actually, by definition, this must be the first
                                       'pattern in the set

        Case Is > 1
            target = t Mod run.SampleSize

    End Select

    Compare (target)                    'identify closest model vector to
                                        'current pattern, by unit number
    For l = 1 To run.SizeOfMap
        weight = NghbrWt((t), winner, (l))
                                        'each unit in the map will have its own
                                        '"neighbourhood" weight indicating how much the
                                        'model vector will be moved in the direction of
                                        'the input pattern.

        For v = 1 To run.NumberOfItems
            layer(l).CycleError(v) = layer(l).CycleError(v) + weight *
                layer(l).PatError(v)
            layer(l).PatError(v) = 0

                                        'accumulate error for all units into
                                        '.CycleError(v), then reset .PatError(v) for next
                                        'pattern
        Next v
    Next l

'TIME TO UPDATE THE MODEL VECTORS

    If (t Mod run.EpochSize) = 0 Then
                                        'if Epoch (user-entered) is complete, update model
                                        'vectors.
        For l = 1 To run.SizeOfMap
            For v = 1 To run.NumberOfItems
                layer(l).vector(v) = layer(l).vector(v) + (layer(l).CycleError(v) /
                        run.EpochSize)

                                        'NOTE: CycleError has already been scaled by alpha
                                        ' (& radius if GAUSSIAN)
                layer(l).CycleError(v) = 0
                                        'reset .CycleError(v)
            Next v
        Next l
```

```
    End If

Next t                                  'Do next training cycle

TestAllPatterns                         'When training is over, test all patterns
                                        'and create statistics.

End Sub
```

```
Public Sub ScramblePatterns()           'Stolen from Dawson, public stuff

'This routine takes all the input patterns in an epoch and randomizes their order.

Dim v As Integer                        'loop counter
Dim CurIndex As Long
Dim NewIndex As Long
Dim TmpValue As Integer

For CurIndex = 1 To run.SampleSize - 1
    NewIndex = Int(Rnd * (run.SampleSize - CurIndex + 1) + CurIndex)

                                        'Swap the Items.

    For v = 1 To run.NumberOfItems + 1
        TmpValue = patterns(NewIndex, v)
        patterns(NewIndex, v) = patterns(CurIndex, v)
        patterns(CurIndex, v) = TmpValue
    Next v
Next CurIndex

End Sub
```

```
Public Sub Compare(inp As Integer)

'Determines the closest and 2nd closest model vectors to the presented pattern,
'i.e., the winning unit(s).  The winner determines the centre of the neighbourhood,
'and the second and first together allows calculation of topological preservation (TP).

Dim l, v As Integer                     'loop counters
Dim CurrentError() As Double            'total squared error in current pattern
Dim xwin, ywin, xsec, ysec As Long      'co-ordinates of winner and second in the map

ReDim CurrentError(run.SizeOfMap)

'FIRST, find the WINNER (the closest model vector to the input pattern)

winner = 1                              'initialize winning unit to UNIT 1

For l = 1 To run.SizeOfMap
    For v = 1 To run.NumberOfItems
        layer(l).PatError(v) = patterns(inp, v) - layer(l).vector(v)

                                        'note that PatError is the simple 'difference
                                        'between input and model (not 'RMS or ^2) since
                                        'the index "v" stands 'for the element of each
                                        'vector

        CurrentError(l) = CurrentError(l) + layer(l).PatError(v) ^ 2

                                        'This step squares PatError (i.e.,
                                        'distance along dimension v) and adds it
                                        'to the total
    Next v
```

```
        If ((CurrentError(1) ^ 0.5) < (CurrentError(winner) ^ 0.5)) Then
         winner = 1

                                          'after error (distance) is calcuated to
                                          'unit 1, determine if distance is less
                                          'than distance to current winning unit.
                                          'If Y, make new winner.

        End If
Next 1

'SECOND, find the SECOND (the second closest unit to the input vector).

second = 1
If winner = 1 Then second = 2

For 1 = 1 To run.SizeOfMap
    For v = 1 To run.NumberOfItems
        layer(1).PatError(v) = patterns(inp, v) - layer(1).vector(v)
        CurrentError(1) = CurrentError(1) + layer(1).PatError(v) ^ 2
    Next v
        If ((CurrentError(1) ^ 0.5) < (CurrentError(second) ^ 0.5)) And Not (winner = 1)
Then
                second = 1
        End If

Next 1

'THIRD, find the co-ordinates of the WINNER and the SECOND.

xwin = winner Mod 4
ywin = (winner - (winner Mod 4)) / 4

If xwin = 0 Then
    xwin = 4
Else
    ywin = ywin + 1
End If

xsec = second Mod 4
ysec = (second - (second Mod 4)) / 4

If xsec = 0 Then
    xsec = 4
Else
    ysec = ysec + 1
End If

'FOURTH, calculate ADJACENCY.  Note that two diagonally units adjacent qualify.

SOMadj = 0
If (Abs(ywin - ysec) ^ 2 + Abs(xwin - xsec) ^ 2) < 4 Then SOMadj = 1

End Sub
```

```
Public Function NghbrWt(time As Long, winner As Integer, OtherUnit As Integer) As Double

'This function calculates the amount that a particular model vector will be adjusted
'given its proximity to the winning unit.
'Arguments:
'           TIME        cycle number
'           WINNER      the index of the winning unit
'           OTHERUNIT   the index of the unit whose weight is being calculated

Dim StartingRadius, radius As Double          'the radius of the neighbourhood decreases
                                              'over time
                                             'StartingRadius is the maximum radius
```

```
Dim xDist, yDist, distance As Double          'distance in each co-ordinate direction, and
                                              'overall
Dim result As Double                          'placeholder for weight
Dim alpha As Double                           'learning rate, which also decreases over
                                              'time
Dim RadiusType As Integer                     'could be GATE or GAUSSIAN, see const
                                              'declar.


'=============================================================
'LEARNING RATE STUFF
If time < 2000 Then
   alpha = 1 - 0.00048 * time
Else
    alpha = 0.005 + 0.035 - (time - 2000) * (0.035 / run.cycles)
End If
                                              ' From Kohonen (1990), pg 1470

'The following code is for a logistic learning rate
    'alpha = 0.01 + 0.99 * (Exp(5 - (time / 1000)) / (1 + Exp(5 - (time / 1000))))
'=============================================================

xDist = (layer(winner).locationx - layer(OtherUnit).locationx) ^ 2
yDist = (layer(winner).locationy - layer(OtherUnit).locationy) ^ 2

distance = xDist + yDist                      'according to 3.4, p 111, distance is the
                                              'Euclidian norm SQUARED, i.e., the sum of
                                              'xDist and yDist

'=============================================================
'RADIUS OF NEIGHBOURHOOD FUNCTION

RadiusType = GATE
'RadiusType = GAUSSIAN

StartingRadius = (run.XofMap + run.YofMap) / 4
                                              'rule of thumb, Radius should start
                                              'as half of diameter of SOM (p 112)
Select Case RadiusType
                                              'If RadiusType = GATE, the same learning
                                              'rate is applied to the entire
                                              'neighbourhood.
    Case GATE
        If (4 * time < run.cycles) Then
            radius = StartingRadius * (1 - 4 * time / run.cycles)
        Else
            radius = 0                        'Update only the winning unit for the last
                                              'phase
        End If

        If (radius ^ 2 < distance) Then
            result = 0
        Else
            result = alpha
        End If
                                              'If RadiusType = GAUSSIAN, learning rate
                                              'decreases smoothly from the centre of the
                                              'neighbourhood.

    Case GAUSSIAN
        radius = 0.5 + StartingRadius * (Exp(5 - time / 500) / (1 + Exp(5 - time / 500)))
        result = alpha * Exp(-distance / radius ^ 2)

End Select
'=============================================================

NghbrWt = result

End Function
```

```
Public Sub TestAllPatterns()

'The following code outputs the index, the winning unit, and the unit error
'of all input patterns
'=================================================================

Dim t, v, l                As Integer      'loop counters

For t = 1 To run.SampleSize                'This time, we only need to check each
pattern once.

        Compare (t)                        'get the winner and second for each pattern

        For v = 1 To run.NumberOfItems + 1
                                           '+1 so that index is outputted also
                                           'outputs the input data to XL sheet
            Sheet3.Cells(t, v) = patterns(t, v)
        Next v

            Sheet3.Cells(t, (run.NumberOfItems + 2)) = winner
                                           'outputs winning unit
            Sheet3.Cells(t, (run.NumberOfItems + 3)) = GetError((t), (winner))
                                           'outputs error of winning unit
        For l = 1 To run.SizeOfMap
                                           'now, get the error for each unit
            Sheet3.Cells(t, (l + run.NumberOfItems + 4)) = GetError((t), (l))
        Next l

        TE = TE + SOMadj                   'This calculates the total number of
                                           'patterns where WINNER and SECOND are
                                           'adjacent.  TP = (TE / run.SampleSize)

Next t

End Sub
```

```
Public Function GetError(PatIndex As Integer, Unitnum As Integer) As Double

'This function returns the Euclidian Distance between the input pattern and the
'model vector associated with the unit indexed by unitnum.

Dim v As Integer                          'loop counter
Dim Error, CurrentError, result As Double

    For v = 1 To run.NumberOfItems
        CurrentError = (patterns(PatIndex, v) - layer(Unitnum).vector(v)) ^ 2
        Error = Error + CurrentError
    Next v
                                           'Error is the SS error for each model
                                           'pattern with the given input pattern

result = Error ^ 0.5

GetError = result

End Function
```

```
Public Sub CalculateMeasures()

'This code determines the classification agreement, ratios of variance within and between
'clusters, mean QE

Dim WithinGroupSS() As Double              'Variance of all patterns assigned to a
                                           'given unit
Dim AvgWGSS, WeightedAvgWGSS As Double     'Average of WithinGroupVar across all 16
                                           'units, non-weighted and weighted by number
                                           'of patterns
Dim GroupCount() As Integer                'Number of patterns assigned to a given unit
```

```
Dim first(), last() As Integer                  'indexes of all patterns assigned to a
                                                'single unit
Dim r, i, j As Integer                          'counter variables
Dim start As Integer

ReDim WithinGroupSS(run.SizeOfMap), GroupCount(run.SizeOfMap), first(run.SizeOfMap),
last(run.SizeOfMap)

For j = 1 To run.SizeOfMap
    For i = 1 To run.NumberOfItems
        Sheet1.Cells(j, i) = layer(j).vector(i)
                                                'Outputs all Model Vectors to Sheet1
    Next i
Next j

                                                'This sorts all patterns according to WINNER
Sheet3.Activate
Range("A1:AF500").Select
    Range("AF500").Activate
    Selection.sort Key1:=Range("N1"), Order1:=xlAscending, Header:=xlGuess, _
        OrderCustom:=1, MatchCase:=False, Orientation:=xlTopToBottom

GetTrueCCPs
Classify
GetBayes
getdistances

j = 1
first(j) = 1
For r = 1 To run.SampleSize
    Sheet3.Cells(r, 16) = Sheet3.Cells(r, 15) ^ 2
                                                'square the euclidian distance as a step to
                                                'get SS within cluster
    If Cells(r, 14) > Cells(first(j), 14) Then
                                                'Find the place where next unit
                                                'takes over
        last(j) = r - 1
        GroupCount(j) = last(j) - first(j) + 1
        j = j + 1
        first(j) = r
    End If
    If first(j) > 1 Then
        If Cells(r, 14) = Cells(first(j - 1), 14) + 2 Then
            first(j) = 999
            last(j) = 999
            GroupCount(j) = 0
            j = j + 1
            first(j) = r
        End If
    End If
Next r

last(run.SizeOfMap) = run.SampleSize
GroupCount(run.SizeOfMap) = last(run.SizeOfMap) - first(run.SizeOfMap) + 1

Sheet2.Cells(18, 1) = WorksheetFunction.Average(Range(Sheet3.Cells(1, 15),
Sheet3.Cells(run.SampleSize, 15)))
Sheet2.Cells(18, 2) = TE / run.SampleSize
                                                'Puts mean quantization error on Row 18,
                                                'Column 1 of Sheet2
                                                'Puts topological error on row 18, col 2 of
                                                'sheet2
For j = 1 To run.SizeOfMap
    If first(j) = last(j) Then  'first(j) = last(j) means cluster with n=(0, 1)
        WithinGroupSS(j) = 0
    Else
        WithinGroupSS(j) = WorksheetFunction.Sum(Range(Sheet3.Cells(first(j), 16),
Sheet3.Cells(last(j), 16)))
                                                'this gets us within gp SS, necessary for
                                                'the Calinski & Harabasz (1974) index
    End If
```

```
    Sheet2.Cells(j, 3) = WithinGroupSS(j)
    Sheet2.Cells(j, 4) = GroupCount(j)
    AvgWGSS = AvgWGSS + WithinGroupSS(j)
    WeightedAvgWGSS = WeightedAvgWGSS + WithinGroupSS(j) * GroupCount(j)
Next j

AvgWGSS = AvgWGSS / (run.SampleSize - run.SizeOfMap)
WeightedAvgWGSS = WeightedAvgWGSS / run.SampleSize

Sheet2.Cells(17, 7) = AvgWGSS
Sheet2.Cells(18, 4) = WeightedAvgWGSS
                                        'Avg within group and avg weighted within
                                        'group variance outputted here
For j = 1 To run.NumberOfItems
    Sheet1.Cells(18, j) = WorksheetFunction.Average(Range(Sheet1.Cells(1, j),
Sheet1.Cells(16, j)))
Next j
                                        'this code calculates the centroid of the
                                        'map

For i = 1 To run.SizeOfMap
    Sheet1.Cells(i, 14) = WorksheetFunction.SumXMY2(Range(Sheet1.Cells(18, 1), _
            Sheet1.Cells(18, 12)), Range(Sheet1.Cells(i, 1), Sheet1.Cells(i, 12)))
Next i
                                        'this code calculates the distance between
                                        'the centroid and each individual point

Sheet2.Cells(17, 6) = WorksheetFunction.Sum(Range(Sheet1.Cells(1, 14), _
            Sheet1.Cells(16, 14))) / (run.SizeOfMap - 1)
                                        'outputted is the variance of each unit to
                                        'the centroid in the map

Sheet2.Cells(18, 6) = Sheet2.Cells(17, 6) / Sheet2.Cells(17, 7)

End Sub
```

```
Public Sub GetTrueCCPs()

'This sub calculates and stores the Class Conditional Item Probabilities

Dim r, class, item As Integer
Dim classsum() As Integer
Dim tempclass As Integer

ReDim classsum(4, run.NumberOfItems)

For r = 1 To run.SampleSize

    Select Case Sheet3.Cells(r, 13)     'This determines the intended classification
                                        'since Class 1 simulees were the first
                                        '125/500, etc.

    Case 1 To Round((run.SampleSize / 4), 0)
        tempclass = 1
    Case Round((run.SampleSize / 4), 0) To Round((run.SampleSize / (4 / 2)), 0)
        tempclass = 2
    Case Round((run.SampleSize / (4 / 2)), 0) To (run.SampleSize _
            - Round((run.SampleSize / 4), 0))
        tempclass = 3
    Case Else
        tempclass = 4

    End Select

    For item = 1 To run.NumberOfItems
        classsum(tempclass, item) = classsum(tempclass, item) + Sheet3.Cells(r, item)
    Next item
                                        'Determines the number of examinees in each
                                        'class that got each item correct
```

```
    Sheet3.Cells(r, 35) = tempclass          'Column AI gets SIM classification

Next r

For class = 1 To 4
    For item = 1 To run.NumberOfItems
        Sheet1.Cells(class + 20, item) = classsum(class, item) / 125
    Next item
Next class
                                             'Outputs Class Conditional Item Probability
End Sub
```

```
Public Sub Classify()

'This sub gives the SOM "classification" by determining the simulated class most often
'associated with a given SOM unit.

Dim bin(16, 4) As Integer
Dim WinningClass(16) As Integer
Dim row, class, unit As Integer
Dim CU, CC As Integer

For row = 1 To run.SampleSize
    CU = Int(Sheet3.Cells(row, 14))          'WINNER
    CC = Int(Sheet3.Cells(row, 35))          'Simulated Class
    bin(CU, CC) = bin(CU, CC) + 1            'Increment counter for bin indexed by
                                             'WINNER, Simulated Class
Next row

For unit = 1 To 16
    WinningClass(unit) = 1
    For class = 2 To 4
        If bin(unit, class) > bin(unit, WinningClass(unit)) Then WinningClass(unit) =
class
    Next class
Next unit
                                             'Determine Class that won most often for
                                             'given unit.
For row = 1 To run.SampleSize
    Sheet3.Cells(row, 36) = WinningClass(Cells(row, 14))
Next row
                                             'Column AJ gets SOM classification
End Sub
```

```
Public Sub GetBayes()

'This function calculates the BAYESIAN classification, i.e., based on the SOM derived
'CCP's, determines the likelihood of each input pattern being a perturbation of the CGP's
'for each simulated class.

Dim row, class, item, BayesClass As Integer
Dim ClassProb() As Double
Dim CPProd() As Double
Dim mostlikely As Integer
Dim agreement() As Integer

ReDim ClassProb(4, run.NumberOfItems)
ReDim CPProd(4), agreement(3)

For row = 1 To run.SampleSize
    For item = 1 To run.NumberOfItems
                                             'Following code calculates likelihood for
                                             'each input pattern and each class.
        If Sheet3.Cells(row, item) = 1 Then
            For class = 1 To 4
                Sheet1.Cells(40 + class, item) = Sheet1.Cells(20 + class, item)
            Next class
        Else
```

```
            For class = 1 To 4
                Sheet1.Cells(40 + class, item) = 1 - (Sheet1.Cells(20 + class, item))
            Next class
        End If
                                                'Determines most likely class for each input
                                                'pattern
    Next item
    mostlikely = 1
    For class = 1 To 4
        CPProd(class) = WorksheetFunction.Product(Range(Sheet1.Cells(40 + class, 1), _
                        Sheet1.Cells(40 + class, 12)))
        If CPProd(class) > CPProd(mostlikely) Then
            mostlikely = class
        End If
    Next class
    Sheet3.Cells(row, 37) = mostlikely

Next row
                                                'The following code calculates the agreement
                                                'between the three classifications:
                                                'simulated, SOM-derived, and Bayesian.
For row = 1 To run.SampleSize
    If Sheet3.Cells(row, 35) = Sheet3.Cells(row, 36) Then
        agreement(1) = agreement(1) + 1
    End If
    If Sheet3.Cells(row, 35) = Sheet3.Cells(row, 37) Then
        agreement(2) = agreement(2) + 1
    End If
    If Sheet3.Cells(row, 36) = Sheet3.Cells(row, 37) Then
        agreement(3) = agreement(3) + 1
    End If
Next row

Sheet2.Cells(17, 9) = "sim / som"
Sheet2.Cells(18, 9) = agreement(1)
Sheet2.Cells(17, 10) = "sim / bayes"
Sheet2.Cells(18, 10) = agreement(2)
Sheet2.Cells(17, 11) = "som / bayes"
Sheet2.Cells(18, 11) = agreement(3)

End Sub
```

```
Public Sub getdistances()

'This code gets the correlation between distances between model vectors and distances in
'the map

Dim r1, r2 As Integer

For r1 = 1 To run.SizeOfMap
    For r2 = 1 To run.SizeOfMap

'FIRST, calculate distances between model vectors

    Sheet1.Cells((50 + r2), r1) = (WorksheetFunction.SumXMY2(Range(Sheet1.Cells(r1, 1), _
        Sheet1.Cells(r1, run.NumberOfItems)), Range(Sheet1.Cells(r2, 1), _
        Sheet1.Cells(r2, run.NumberOfItems)))) ^ 0.5

'SECOND, calculate distances between corresponding map locations

    Sheet1.Cells(70 + r2, r1) = ((layer(r1).locationx - layer(r2).locationx) ^ 2 + _
        (layer(r1).locationy - layer(r2).locationy) ^ 2) ^ 0.5

    Next r2
Next r1
                                                'Output correlation between distances on
                                                'Sheet2

    Sheet2.Cells(18, 3) = WorksheetFunction.Correl(Range(Sheet1.Cells(50 + 1, 1), _
        Sheet1.Cells(50 + run.SizeOfMap, run.SizeOfMap)), _
```

```
        Range(Sheet1.Cells(70 + 1, 1), Sheet1.Cells(70 + run.SizeOfMap, _
        run.SizeOfMap)))

End Sub
```

```
Public Sub SaveData()        'Saves various format data files

Dim l, v As Integer
Dim text, XLSFile, TextFile, sumfile  As String

XLSFile = run.FileStem & ".xls"
TextFile = run.FileStem & ".out"
sumfile = Left$(run.FileStem, 9) & ".dat"

Open TextFile For Output As #1
Open sumfile For Append As #2

For l = 1 To run.SizeOfMap
    For v = 1 To run.NumberOfItems

       text = layer(l).vector(v)
        Write #1, text
        text = ""

    Next v
Next l

For l = 1 To 12
    text = text & Str(Sheet2.Cells(18, l))
Next l

Write #2, text

text = ""

ActiveWorkbook.SaveAs (XLSFile)
Close #1
Close #2

End Sub
```